



(43) 國際公開日  
2004 年 11 月 25 日 (25.11.2004)

## PCT

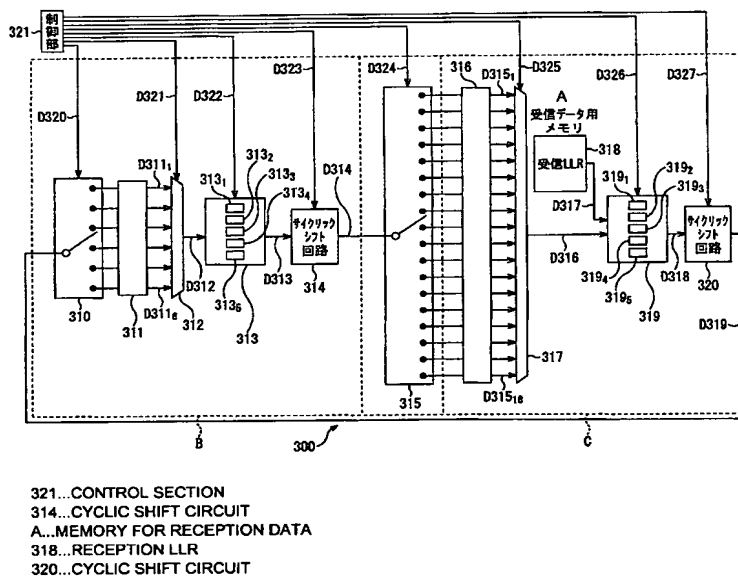
(10) 国際公開番号  
**WO 2004/102811 A1**

- |                                   |   |   |  |
|-----------------------------------|---|---|--|
| (51) 国際特許分類 <sup>7</sup> :        | H03M 13/09, 13/19   | (72) 発明者; および   |  |
| (21) 国際出願番号:                      | PCT/JP2004/005562   | (75) 発明者/出願人 (米国についてのみ):  | 横川 峰志 (YOKOKAWA, Takashi) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 宮内 俊之 (MIYAUCHI, Toshiyuki) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 飯田 康博 (IIDA, Yasuhiro) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). |
| (22) 国際出願日:                       | 2004 年 4 月 19 日 (19.04.2004)  |   |  |
| (25) 国際出願の言語:                     | 日本語   |   |  |
| (26) 国際公開の言語:                     | 日本語   |   |  |
| (30) 優先権データ:                      |   | (74) 代理人: 稻本 義雄 (INAMOTO, Yoshio); 〒1600023 東京都新宿区西新宿 7 丁目 1 1 番 1 8 号 7 1 1 ビルディング 4 階 Tokyo (JP). |  |
| 特願2003-133941                     | 2003 年 5 月 13 日 (13.05.2003)  | JP  |  |
| 特願2003-294383                     | 2003 年 8 月 18 日 (18.08.2003)  | JP  |  |
| (71) 出願人 (米国を除く全ての指定国について):       | ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 Tokyo (JP). |   |  |
| (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): | AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR,                                       |   |  |

〔続葉有〕

- (54) Title:** DECODING DEVICE, DECODING METHOD, AND PROGRAM

- (54) 発明の名称: 復号装置および復号方法、並びにプログラム



- (S7) Abstract:** There are provided a decoding device, a decoding method, and a program for realizing decoding of the LDPC code capable of suppressing the circuit size, suppressing the operation frequency to a sufficiently realizable range, and easily controlling memory access. The inspection matrix of the LDPC code is composed of a combination of a unit matrix  $p \times p$ , the unit matrix in which one or more 1 have become 0, their cyclic shifts, a sum of them, and a 0 matrix of  $p \times p$ . A check node calculation section (313) simultaneously performs P check node calculations while a variable node calculation section (319) simultaneously performs P variable node calculations.

- (57) 要約: 本発明は、回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことができるLDPC符号の復号を実現する復号装置および復号方法、並びにプログラムに関する。LDPC符号の検査行列は、 $P \times P$ の単位行列、その単位行列の1のうちの1個から数個が0になった行列、それらのサイクリックシフト、それ

[統葉有]

**WO 2004/102811 A1**



BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

KZ, MD, RU, TJ, TM), ヨーロッパ(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告書

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG,

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

## 明細書

## 復号装置および復号方法、並びにプログラム

## 技術分野

- 5 本発明は、復号装置および復号方法、並びにプログラムに関し、特に、低密度パリティ検査符号（LDPC 符号）による符号化が施された符号の復号を行う復号装置および復号方法、並びにプログラムに関する。

## 背景技術

- 10 近年、例えば、移動体通信や深宇宙通信といった通信分野、及び地上波又は衛星デジタル放送といった放送分野の研究が著しく進められているが、それに伴い、誤り訂正符号化及び復号の効率化を目的として符号理論に関する研究も盛んに行われている。

- 符号性能の理論的限界としては、いわゆるシャノン(C. E. Shannon)の通信  
15 路符号化定理によって与えられるシャノン限界が知られている。符号理論に関する研究は、このシャノン限界に近い性能を示す符号を開発することを目的として行われている。近年では、シャノン限界に近い性能を示す符号化方法として、例えば、並列接続畳み込み符号(PCCC(Parallel Concatenated Convolutional Codes))や、縦列接続畳み込み符号(SCCC(Serially Concatenated  
20 Convolutional Codes))といった、いわゆるターボ符号化(Turbo coding)と呼ばれる手法が開発されている。また、これらのターボ符号が開発される一方で、古くから知られる符号化方法である低密度パリティ検査符号(Low Density Parity Check codes)（以下、LDPC 符号という）が脚光を浴びつつある。

- LDPC 符号は、R. G. Gallager による「R. G. Gallager, "Low  
25 Density Parity Check Codes", Cambridge, Massachusetts: M. I. T. Press, 1963」において最初に提案されたものであり、その後、「D. J. C. MacKay, "Good error correcting codes based on very sparse

matrices", Submitted to IEEE Trans. Inf. Theory, IT-45, pp. 399-431, 1999] や、[M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. A. Spielman, "Analysis of low density codes and improved designs using irregular graphs", in  
5 Proceedings of ACM Symposium on Theory of Computing, pp. 249-258, 1998] 等において再注目されるに至ったものである。

LDPC 符号は、近年の研究により、ターボ符号等と同様に、符号長を長くしていくにしたがって、シャノン限界に近い性能が得られることがわかりつつある。また、LDPC 符号は、最小距離が符号長に比例するという性質があることから、  
10 その特徴として、ブロック誤り確率特性がよく、さらに、ターボ符号等の復号特性において観測される、いわゆるエラーフロア現象が殆ど生じないことも利点として挙げられる。

以下、このような LDPC 符号について具体的に説明する。なお、LDPC 符号は、線形符号であり、必ずしも 2 元である必要はないが、ここでは、2 元であるもの  
15 として説明する。

LDPC 符号は、その LDPC 符号を定義する検査行列 (parity check matrix) が疎なものであることを最大の特徴とするものである。ここで、疎な行列とは、行列のコンポーネントの "1" の個数が非常に少なく構成されるものであり、疎な検査行列を  $H$  で表すものとする、そのような検査行列としては、例えば、図 1  
20 に示すように、各列のハミング重み ("1" の数) (weight) が "3" であり、且つ、各行のハミング重みが "6" であるもの等がある。

このように、各行及び各列のハミング重みが一定である検査行列  $H$  によって定義される LDPC 符号は、レギュラー LDPC 符号と称される。一方、各行及び各列のハミング重みが一定でない検査行列  $H$  によって定義される LDPC 符号は、イレ  
25 ギュラー LDPC 符号と称される。

このような LDPC 符号による符号化は、検査行列  $H$  に基づいて生成行列  $G$  を生成し、この生成行列  $G$  を 2 元の情報メッセージに対して乗算することによって

符号語を生成することで実現される。具体的には、LDPC 符号による符号化を行う符号化装置は、まず、検査行列  $H$  の転置行列  $H^T$  との間に、式  $GH^T=0$  が成立する生成行列  $G$  を算出する。ここで、生成行列  $G$  が、 $k \times n$  行列である場合には、符号化装置は、生成行列  $G$  に対して  $k$  ビットからなる情報メッセージ（ベクトル  $u$ ）を乗算し、 $n$  ビットからなる符号語  $c$  ( $=uG$ ) を生成する。この符号化装置によって生成された符号語は、値が“0”の符号ビットが“+1”に、値が“1”の符号ビットが“-1”にといったようにマッピングされて送信され、所定の通信路を介して受信側において受信されることになる。

一方、LDPC 符号の復号は、Gallager が確率復号 (Probabilistic Decoding) と称して提案したアルゴリズムであって、バリアブルノード (variable node (メッセージノード (message node) と呼ばれる。)) と、チェックノード (check node) とからなる、いわゆるタナーグラフ (Tanner graph) 上での確率伝播 (belief propagation) によるメッセージ・パッシング・アルゴリズムによって行うことが可能である。ここで、以下、適宜、バリアブルノードとチェックノードを、単に、ノードともいう。

しかしながら、確率復号においては、各ノード間で受け渡されるメッセージが実数値であることから、解析的に解くためには、連続した値をとるメッセージの確率分布そのものを追跡する必要があり、非常に困難を伴う解析を必要とすることになる。そこで、Gallager は、LDPC 符号の復号アルゴリズムとして、アルゴリズム A 又はアルゴリズム B を提案している。

LDPC 符号の復号は、一般的には、図 2 に示すような手順にしたがって行われる。なお、ここでは、受信値を  $U_0(u_{0i})$  とし、チェックノードから出力されるメッセージを  $u_j$  とし、バリアブルノードから出力されるメッセージを  $v_i$  とする。また、ここでは、メッセージとは、値の“0”らしさを、いわゆる対数尤度比 (log likelihood ratio) で表現した実数値である。

まず、LDPC 符号の復号においては、図 2 に示すように、ステップ S 1 1 において、受信値  $U_0(u_{0i})$  が受信され、メッセージ  $u_j$  が“0”に初期化されるとともに、

繰り返し処理のカウンタとしての整数をとる変数  $k$  が“0”に初期化され、ステップ S 1 2 に進む。ステップ S 1 2 において、受信値  $U_0(u_{0i})$  に基づいて、式

(1) に示す演算を行うことによってメッセージ  $v_i$  が求められ、さらに、このメッセージ  $v_i$  に基づいて、式 (2) に示す演算を行うことによってメッセージ  $u_j$  が求められる。

$$v_i = u_{0i} + \sum_{j=1}^{d_v-1} u_j \quad \dots (1)$$

$$\tanh\left(\frac{u_j}{2}\right) = \prod_{i=1}^{d_c-1} \tanh\left(\frac{v_i}{2}\right) \quad \dots (2)$$

ここで、式 (1) と式 (2) における  $d_v$  と  $d_c$  は、それぞれ、検査行列  $H$  の縦方向（行方向）と横方向（列方向）の“1”の個数を示す任意に選択可能とされるパラメータであり、例えば、(3,6)符号の場合には、 $d_v=3$ 、 $d_c=6$  となる。

なお、式 (1) または (2) の演算においては、それぞれ、メッセージを出力しようとする枝(edge)から入力されたメッセージを、和または積演算のパラメータとしては用いないことから、和または積演算の範囲が、1 乃至  $d_v-1$  または 1 乃至  $d_c-1$  となっている。また、式 (2) に示す演算は、実際には、2 入力  $v_1$ ,  $v_2$  に対する 1 出力で定義される式 (3) に示す関数  $R(v_1, v_2)$  のテーブルを予め作成しておき、これを式 (4) に示すように連続的（再帰的）に用いることによって行われる。

$$x = 2 \tanh^{-1} \{ \tanh(v_1/2) \tanh(v_2/2) \} = R(v_1, v_2) \quad \dots (3)$$

$$u_j = R(v_1, R(v_2, R(v_3, \dots R(v_{d_c-2}, v_{d_c-1}))) \quad \dots (4)$$

ステップ S 1 2 では、さらに、変数  $k$  が“1”だけインクリメントされ、ステップ S 1 3 に進む。ステップ S 1 3 では、変数  $k$  が所定の繰り返し復号回数  $N$  以上であるか否かが判定される。ステップ S 1 3 において、変数  $k$  が  $N$  以上ではないと判定された場合、ステップ S 1 2 に戻り、以下、同様の処理が繰り返され

る。

また、ステップ S 1 3 において、変数  $k$  が  $N$  以上であると判定された場合、ステップ S 1 4 に進み、式 (5) に示す演算を行うことによって最終的に出力する復号結果としてのメッセージ  $v$  が求められて出力され、LDPC 符号の復号処理が終了する。

$$v = u_{0i} + \sum_{j=1}^{d_v} u_j \quad \dots (5)$$

ここで、式 (5) の演算は、式 (1) の演算とは異なり、バリエブルノードに接続している全ての枝からの入力メッセージを用いて行われる。

このような LDPC 符号の復号は、例えば (3, 6) 符号の場合には、図 3 に示すように、各ノード間でメッセージの授受が行われる。なお、図 3 における "=" で示すノード (バリエブルノード) では、式 (1) に示した演算が行われ、"+" で示すノード (チェックノード) では、式 (2) に示した演算が行われる。特に、アルゴリズム A においては、メッセージを 2 元化し、"+" で示すノードにて、 $d_v - 1$  個の入力メッセージの排他的論理和演算を行い、 "=" で示すノードにて、受信値  $R$  に対して、 $d_v - 1$  個の入力メッセージが全て異なるビット値であった場合には、符号を反転して出力する。

また、一方で、近年、LDPC 符号の復号の実装法に関する研究も行われている。実装方法について述べる前に、まず、LDPC 符号の復号を模式化して説明する。

図 4 は、(3, 6) LDPC 符号 (符号化率 1/2、符号長 12) の検査行列 (parity check matrix) の例である。LDPC 符号の検査行列は、図 5 のように、タナグラフを用いて書き表すことができる。ここで、図 5 において、"+" で表わされるのが、チェックノードであり、 "=" で表わされるのが、バリエブルノードである。チェックノードとバリエブルノードは、それぞれ、検査行列の行と列に対応する。チェックノードとバリエブルノードとの間の結線は、枝 (edge) であり、検査行列の "1" に相当する。即ち、検査行列の第  $j$  行第  $i$  列のコンポーネントが 1 である場合には、図 5 において、上から  $i$  番目のバリエブルノード ( "=" のノード)

と、上から  $j$  番目のチェックノード（“+”のノード）とが、枝により接続される。枝は、バリエブルノードに対応する符号ビットが、チェックノードに対応する拘束条件を持つことを表わす。なお、図 5 は、図 4 の検査行列のタナーグラフとなっている。

- 5 LDPC 符号の復号方法であるサンプロダクトアルゴリズム (Sum Product Algorithm) は、バリエブルノードの演算とチェックノードの演算とを繰り返す行う。

- バリエブルノードでは、図 6 のように、式 (1) の演算を行う。すなわち、図 6 において、計算しようとしている枝に対応するメッセージ  $v_i$  は、バリエブル  
10 ノードに繋がっている残りの枝からのメッセージ  $u_1$  および  $u_2$  と、受信情報  $u_{0i}$  を用いて計算される。他の枝に対応するメッセージも同様に計算される。

- チェックノードの演算について説明する前に、式 (2) を、式  
 $a \times b = \exp\{\ln(|a|) + \ln(|b|)\} \times \text{sign}(a) \times \text{sign}(b)$  の関係を用いて、式 (6) の  
15 ように書き直す。但し、 $\text{sign}(x)$  は、 $x \geq 0$  のとき 1 であり、 $x < 0$  のとき -1 である。

$$\begin{aligned}
 u_j &= 2 \tanh^{-1} \left( \prod_{i=1}^{d_c-1} \tanh \left( \frac{v_i}{2} \right) \right) \\
 &= 2 \tanh^{-1} \left[ \exp \left\{ \sum_{i=1}^{d_c-1} \ln \left( \left| \tanh \left( \frac{v_i}{2} \right) \right| \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign} \left( \tanh \left( \frac{v_i}{2} \right) \right) \right] \\
 &= 2 \tanh^{-1} \left[ \exp \left\{ - \left( \sum_{i=1}^{d_c-1} - \ln \left( \tanh \left( \frac{|v_i|}{2} \right) \right) \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \right] \\
 &\quad \dots (6)
 \end{aligned}$$

- 更に、 $x \geq 0$  において、 $\phi(x) = \ln(\tanh(x/2))$  と定義すると、 $\phi^{-1}(x) = 2 \tanh^{-1}(e^{-x})$  であるから、式 (6) は、式 (7) のように書くことができる。  
20



$$u_j = \phi^{-1} \left( \sum_{i=1}^{d_o-1} \phi(|v_i|) \right) \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \quad \dots (7)$$

チェックノードでは、図7のように、式(7)の演算を行う。すなわち、図7において、計算しようとしている枝に対応するメッセージ  $u_j$  は、チェックノードに繋がっている残りの枝からのメッセージ  $v_1, v_2, v_3, v_4, v_5$  を用いて計算される。

5 他の枝に対応するメッセージも同様に計算される。

なお、関数  $\phi(x)$  は、 $\phi(x) = \ln((e^x+1)/(e^x-1))$  と表すことができ、 $x > 0$  において、 $\phi(x) = \phi^{-1}(x)$  である。関数  $\phi(x)$  および  $\phi^{-1}(x)$  をハードウェアに実装する際には、LUT(Look Up Table)を用いて実装される場合があるが、両者共に同一のLUTとなる。

10 サンプロダクトアルゴリズムをハードウェアに実装する場合、式(1)で表わされるバリアブルノード演算および式(7)で表わされるチェックノード演算とを、適度な回路規模と動作周波数で繰り返し行うことが必要である。

復号装置の実装の例として、まず、単純に各ノードの演算を一つずつ順次行うことによって復号を行う場合(full serial decoding)の実装法について説明

15 する。

なお、ここでは、例えば、図8の、30(行)×90(列)の検査行列で表現される符号(符号化率2/3、符号長90)を復号することとする。図8の検査行列の1の数は269であり、従って、そのタナーグラフでは、枝の数は269個となる。ここで、図8の検査行列では、0を、“.”で表現している。

20 図9は、LDPC符号の1回復号を行う復号装置の構成例を示している。

図9の復号装置では、その動作する1クロック(clock)ごとに、1つの枝に対応するメッセージが計算される。

即ち、図9の復号装置は、2つの枝用メモリ100および102、1つのチェックノード計算器101、1つのバリアブルノード計算器103、1つの受信用

25 メモリ104、1つの制御部105からなる。

図9の復号装置では、枝用メモリ100または102からメッセージデータが

1 つずつ読み出され、そのメッセージデータを用いて、所望の枝に対応するメッセージデータが計算される。そして、その計算によって求められたメッセージデータが 1 つずつ後段の枝用メモリ 102 または 100 に格納されていく。繰り返し復号を行う際には、この 1 回復号を行う図 9 の復号装置を複数個縦列に接続するか、もしくは図 9 の復号装置を繰り返し用いることによって、繰り返し復号を実現する。なお、ここでは、例えば、図 9 の復号装置が複数個接続されているものとする。

枝用メモリ 100 は、前段の復号装置（図示せず）のバリエブルノード計算器 103 から供給されるメッセージ D100 を、後段のチェックノード計算器 101 が読み出す順番に格納していく。そして、枝用メモリ 100 は、チェックノード計算のフェーズでは、メッセージ D100 を、格納してある順番通りに、メッセージ D101 として、チェックノード計算器 101 に供給する。

チェックノード計算器 101 は、制御部 105 から供給される制御信号 D106 に基づき、枝用メモリ 100 から供給されるメッセージ D101 を用いて、式（7）に従って演算を行い、その演算によって求められたメッセージ D102 を、後段の枝用メモリ 102 に供給する。

枝用メモリ 102 は、前段のチェックノード計算器 101 から供給されるメッセージ D102 を、後段のバリエブルノード計算器 103 が読み出す順番に格納していく。そして、枝用メモリ 102 は、バリエブルノード計算のフェーズでは、メッセージ D102 を、格納してある順番通りに、メッセージ D103 として、バリエブルノード計算器 103 に供給する。

さらに、バリエブルノード計算器 103 には、制御部 105 から制御信号 D107 が供給されるとともに、受信用メモリ 104 から受信データ D104 が供給される。バリエブルノード計算器 103 は、制御信号 D107 に基づき、枝用メモリ 100 から供給されるメッセージ D103 と受信用メモリ 104 から供給される受信データ D104 を用い、式（1）に従って演算を行い、その演算の結果得られるメッセージ D105 を、図示せぬ後段の復号装置の枝用メモリ 100 に供給する。

受信用メモリ 104 には、LDPC 符号化された受信データ (LDPC 符号) が格納される。制御部 105 は、バリアブルノード演算を制御する制御信号 D106 と、チェックノード演算を制御する制御信号 D107 を、それぞれチェックノード計算器 101 とバリアブルノード計算器 103 に供給する。制御部 105 は、枝用メモリ 100 に全ての枝のメッセージが格納されたとき、チェックノード計算器 101 に制御信号 D106 を供給し、枝用メモリ 102 に全ての枝のメッセージが格納されたとき、バリアブルノード計算器 103 に制御信号 D107 を供給する。

図 10 は、チェックノード演算を 1 つずつ行う図 9 のチェックノード計算器 101 の構成例を示している。

なお、図 10 では、各メッセージが符号ビットを合わせて合計 6 ビット (bit) に量子化されているものとして、チェックノード計算器 101 を表している。また、図 10 では、図 8 の検査行列で表わされる LDPC 符号のチェックノード演算が行われる。さらに、図 10 のチェックノード演算器 101 には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

図 10 のチェックノード計算器 101 は、制御部 105 から供給される、例えば、1 ビットの制御信号 D106 に基づき、枝用メモリ 100 から 1 つずつ読み込まれるメッセージ D101 を用いて、式 (7) にしたがって演算を行う。

即ち、チェックノード計算器 101 では、検査行列の各列に対応するバリアブルノードからの 6 ビットのメッセージ D101 (メッセージ  $v_i$ ) が 1 つずつ読み込まれ、その下位ビットである絶対値 D122 ( $|v_i|$ ) が LUT 121 に、その最上位ビットである符号ビット D121 が EXOR 回路 129 と FIFO (First In First Out) メモリ 133 にそれぞれ供給される。また、チェックノード計算器 101 には、制御部 105 から制御信号 D106 が供給され、その制御信号 D106 は、セレクト 124 とセレクト 131 に供給される。

LUT 121 は、絶対値 D122 ( $|v_i|$ ) に対して、式 (7) における  $\phi(|v_i|)$  の演算を行った 5 ビットの演算結果 D123 ( $\phi(|v_i|)$ ) を読み出し、加算器 122 と

FIFO メモリ 1 2 7 に供給する。

加算器 1 2 2 は、演算結果 D123 ( $\phi(|v_i|)$ ) とレジスタ 1 2 3 に格納されている 9 ビットの値 D124 とを加算することにより、演算結果 D123 を積算し、その結果得られる 9 ビットの積算値をレジスタ 1 2 3 に再格納する。なお、検査行  
5 列の 1 行に亘る全ての枝からのメッセージ D101 の絶対値 D122 ( $|v_i|$ ) に対する演算結果が積算された場合、レジスタ 1 2 3 はリセットされる。

検査行列の 1 行に亘るメッセージ D101 が 1 つずつ読み込まれ、レジスタ 1 2 3 に 1 行分の演算結果 D123 が積算された積算値が格納された場合、制御部 1 0 5 から供給される制御信号 D106 は、0 から 1 に変化する。例えば、行の重み  
10 (row weight) が「9」である場合、制御信号 D106 は、1 から 8 クロック目までは、「0」となり、9 クロック目では「1」となる。

制御信号 D106 が「1」の場合、セクタ 1 2 4 は、レジスタ 1 2 3 に格納されている値、即ち、検査行列の 1 行に亘る全ての枝からのメッセージ D101 (メッセージ  $v_i$ ) から求められた  $\phi(|v_i|)$  が積算された 9 ビットの値 D124 ( $i = 1$   
15 から  $i = d_0$  までの  $\sum \phi(|v_i|)$ ) を選択し、値 D125 として、レジスタ 1 2 5 に出力して格納させる。レジスタ 1 2 5 は、格納している値 D125 を、9 ビットの値 D126 として、セクタ 1 2 4 と加算器 1 2 6 に供給する。制御信号 D106 が「0」の場合、セクタ 1 2 4 は、レジスタ 1 2 5 から供給された値 D126 を選択し、レジスタ 1 2 5 に出力して再格納させる。即ち、検査行列の 1 行に亘る全  
20 ての枝からのメッセージ D101 (メッセージ  $v_i$ ) から求められた  $\phi(|v_i|)$  が積算されるまで、レジスタ 1 2 5 は、前回積算された  $\phi(|v_i|)$  を、セクタ 1 2 4 と加算器 1 2 6 に供給する。

一方、FIFO メモリ 1 2 7 は、レジスタ 1 2 5 から新たな値 D126 ( $i = 1$  から  $i = d_0$  までの  $\sum \phi(|v_i|)$ ) が出力されるまでの間、LUT 1 2 1 が出力した演算  
25 結果 D123 ( $\phi(|v_i|)$ ) を遅延し、5 ビットの値 D127 として減算器 1 2 6 に供給する。減算器 1 2 6 は、レジスタ 1 2 5 から供給された値 D126 から、FIFO メモリ 1 2 7 から供給された値 D127 を減算し、その減算結果を、5 ビットの減算値

D128 として LUT 1 2 8 に供給する。即ち、減算器 1 2 6 は、検査行列の 1 行に亘る全ての枝からのメッセージ D101 (メッセージ  $v_i$ ) から求められた  $\phi(|v_i|)$  の積算値から、求めたい枝からのメッセージ D101 (メッセージ  $v_i$ ) から求められた  $\phi(|v_i|)$  を減算して、その減算値 ( $i = 1$  から  $i = d_c - 1$  までの

5  $\Sigma \phi(|v_i|)$ ) を減算値 D128 として LUT 1 2 8 に供給する。

LUT 1 2 8 は、減算値 D128 ( $i = 1$  から  $i = d_c - 1$  までの  $\Sigma \phi(|v_i|)$ ) に対して、式 (7) における  $\phi^{-1}(\Sigma \phi(|v_i|))$  の演算を行った 5 ビットの演算結果 D129 ( $\phi^{-1}(\Sigma \phi(|v_i|))$ ) を出力する。

以上の処理と並行して、EXOR 回路 1 2 9 は、レジスタ 1 3 0 に格納されている 1 ビットの値 D131 と符号ビット D121 との排他的論理和を演算することにより、符号ビットどうしの乗算を行い、1 ビットの乗算結果 D130 をレジスタ 1 3 0 に再格納する。なお、検査行列の 1 行に亘る全ての枝からのメッセージ D101 の符号ビット D121 が乗算された場合、レジスタ 1 3 0 はリセットされる。

検査行列の 1 行に亘る全ての枝からのメッセージ D101 の符号ビット D121 が乗算された乗算結果 D130 ( $i = 1$  から  $d_c$  までの  $\Pi \text{sign}(v_i)$ ) がレジスタ 1 3 0 に格納された場合、制御部 1 0 5 から供給される制御信号 D106 は、「0」から「1」に変化する。

制御信号 D106 が「1」の場合、セクタ 1 3 1 は、レジスタ 1 3 0 に格納されている値、即ち、検査行列の 1 行に亘る全ての枝からのメッセージ D101 の符号ビット D121 が乗算された値 D131 ( $i = 1$  から  $i = d_c$  までの  $\Pi \text{sign}(v_i)$ ) を選択し、1 ビットの値 D132 としてレジスタ 1 3 2 に出力して格納させる。レジスタ 1 3 2 は、格納している値 D132 を、1 ビットの値 D133 としてセクタ 1 3 1 と EXOR 回路 1 3 4 に供給する。制御信号 D106 が「0」の場合、セクタ 1 3 1 は、レジスタ 1 3 2 から供給された値 D133 を選択し、レジスタ 1 3 2 に出力して再格納させる。即ち、検査行列の 1 行に亘る全ての枝からのメッセージ D101 (メッセージ  $v_i$ ) の符号ビット D121 が乗算されるまで、レジスタ 1 3 2 は、前回格納した値を、セクタ 1 3 1 と EXOR 回路 1 3 4 に供給する。

一方、FIFOメモリ133は、レジスタ132から新たな値D133 ( $i = 1$  から  $i = d_c$  までの  $\Pi \text{sign}(v_i)$ ) が EXOR 回路134に供給されるまでの間、符号ビットD121を遅延し、1ビットの値D134としてEXOR回路134に供給する。

EXOR回路134は、レジスタ132から供給された値D133と、FIFOメモリ133から供給された値D134との排他的論理和を演算することにより、値D133を、値D134で除算し、1ビットの除算結果を除算値D135として出力する。即ち、EXOR回路134は、検査行列の1行に亘る全ての枝からのメッセージD101の符号ビットD121 ( $\text{sign}(|v_i|)$ ) の乗算値を、求めたい枝からのメッセージD101の符号ビットD121 ( $\text{sign}(|v_i|)$ ) で除算して、その除算値 ( $i = 1$  から  $i = d_c - 1$  までの  $\Pi \text{sign}(|v_i|)$ ) を除算値D135として出力する。

チェックノード計算器101では、LUT128から出力された5ビットの演算結果D129を下位5ビットとするとともに、EXOR回路134から出力された1ビットの除算値D135を最上位ビットとする合計6ビットがメッセージD102 (メッセージ  $u_j$ ) として出力される。

以上のように、チェックノード計算器101では、式(7)の演算が行われ、メッセージ  $u_j$  が求められる。

なお、図8の検査行列の行の重みの最大は9であるため、即ち、チェックノードに供給されるメッセージの最大数は9であるため、チェックノード計算器101は、9個のメッセージ ( $\phi(|v_i|)$ ) を遅延させるFIFOメモリ127とFIFOメモリ133を有している。行の重みが9未満の行のメッセージを計算するときには、FIFOメモリ127とFIFOメモリ133における遅延量が、その行の重みの値に減らされる。

図11は、バリアブルノード演算を1つずつ行う図9のバリアブルノード計算器103の構成例を示している。

なお、図11では、各メッセージが符号ビットを合わせて合計6ビット(bit)に量子化されているものとして、バリアブルノード計算器103を表している。また、図11では、図8の検査行列で表わされるLDPC符号のバリアブルノード

演算が行われる。さらに、図 11 のバリアブルノード計算機 103 には、クロック  $ck$  が供給され、クロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

図 11 のバリアブルノード計算器 103 は、制御部 105 から供給される、例えば、1 ビットの制御信号  $D107$  に基づき、枝用メモリ 102 から 1 つずつ読み込まれるメッセージ  $D103$  と、受信用メモリ 104 から読み込まれる受信データ  $D104$  を用いて、式 (1) にしたがって演算を行う。

即ち、バリアブルノード計算器 103 では、検査行列の各行に対応するチェックノードからの 6 ビットのメッセージ  $D103$  (メッセージ  $u_j$ ) が 1 つずつ読み込まれ、そのメッセージ  $D103$  が、加算器 151 と FIFO メモリ 155 に供給される。また、バリアブルノード計算器 103 では、受信用メモリ 104 から 6 ビットの受信データ  $D104$  が 1 つずつ読み込まれ、加算器 156 に供給される。さらに、バリアブルノード計算器 103 には、制御部 105 から制御信号  $D107$  が供給され、その制御信号  $D107$  は、セクタ 153 に供給される。

加算器 151 は、メッセージ  $D103$  (メッセージ  $u_j$ ) とレジスタ 152 に格納されている 9 ビットの値  $D151$  とを加算することにより、メッセージ  $D103$  を積算し、その結果得られる 9 ビットの積算値を、レジスタ 152 に再格納する。なお、検査行列の 1 列に亘る全ての枝からのメッセージ  $D103$  が積算された場合、レジスタ 152 はリセットされる。

検査行列の 1 列に亘るメッセージ  $D103$  が 1 つずつ読み込まれ、レジスタ 152 に 1 列分のメッセージ  $D103$  が積算された値が格納された場合、制御部 105 から供給される制御信号  $D107$  は、「0」から「1」に変化する。例えば、列の重みが「5」である場合、制御信号  $D107$  は、1 から 4 クロック目までは「0」となり、5 クロック目では「1」となる。

制御信号  $D107$  が「1」の場合、セクタ 153 は、レジスタ 152 に格納されている値、即ち、検査行列の 1 列に亘る全ての枝からのメッセージ  $D103$  (メッセージ  $u_j$ ) が積算された 9 ビットの値  $D151$  ( $j = 1$  から  $d_v$  までの  $\sum u_j$ ) を選

択し、レジスタ 154 に出力して格納させる。レジスタ 154 は、格納している値 D151 を、9 ビットの値 D152 として、セクタ 153 と加減算器 156 に供給する。制御信号 D107 が「0」の場合、セクタ 153 は、レジスタ 154 から供給された値 D152 を選択し、レジスタ 154 に出力し再格納させる。即ち、

5 検査行列の 1 列に亘る全ての枝からのメッセージ D103 (メッセージ  $u_j$ ) が積算されるまで、レジスタ 154 は、前回積算された値を、セクタ 153 と加減算器 156 に供給する。

一方、FIFO メモリ 155 は、レジスタ 154 から新たな値 D152 ( $j = 1$  から  $d_v$  までの  $\sum u_j$ ) が出力されるまでの間、チェックノードからのメッセージ D103

10 を遅延し、6 ビットの値 D153 として加減算器 156 に供給する。加減算器 156 は、レジスタ 154 から供給された値 D152 から、FIFO メモリ 155 から供給された値 D153 を減算する。即ち、加減算器 156 は、検査行列の 1 列に亘る全ての枝からのメッセージ D103 (メッセージ  $u_j$ ) の積算値から、求めたい枝からのメッセージ  $u_j$  を減算して、その減算値 ( $j = 1$  から  $d_v - 1$  までの  $\sum u_j$ ) を求

15 める。さらに、加減算器 156 には、その減算値 ( $j = 1$  から  $d_v - 1$  までの  $\sum u_j$ ) に、受信用メモリ 104 から供給された受信データ D104 を加算して、その結果得られる 6 ビットの値をメッセージ D105 (メッセージ  $v_i$ ) として出力する。

以上のように、バリアブルノード計算器 103 では、式 (1) の演算が行われ、

20 メッセージ  $v_i$  が求められる。

なお、図 8 の検査行列の列の重みの最大は 5 であるため、即ち、バリアブルノードに供給されるメッセージの最大数は 5 であるため、バリアブルノード計算器 103 は、5 個のメッセージ ( $u_j$ ) を遅延させる FIFO メモリ 155 を有している。列の重みが 5 未満の列のメッセージを計算するときには、FIFO メモリ 15

25 5 における遅延量が、その列の重みの値に減らされる。

図 9 の復号装置では、検査行列の重みにしたがって、制御部 105 から制御信号が与えられる。そして、図 9 の復号装置によれば、枝用メモリ 100 および 1



02、並びにチェックノード計算器101およびバリエブルノード計算器103のFIFOメモリ127, 133, 155の容量さえ足りれば、制御信号のみを変えることで様々な検査行列のLDPC符号を復号することができる。

5      なお、図示しないが、図9の復号装置において、復号の最終段においては、式(1)のバリエブルノード演算の代わりに、式(5)の演算が行われ、その演算結果が、最終的な復号結果として出力される。

10      図9の復号装置を繰り返し用いて、LDPC符号を復号する場合には、チェックノード演算とバリエブルノード演算とが交互に行われる。即ち、図9の復号装置では、チェックノード計算器101によるチェックノード演算の結果を用いて、バリエブルノード計算器103によりバリエブルノード演算が行われ、バリエブルノード計算器103によるバリエブルノード演算の結果を用いて、チェックノード計算器101によりチェックノード演算が行われる。

15      従って、269の枝を有する図8の検査行列を用いた1回の復号に、 $269 \times 2 = 538$  クロック(clock)を必要とする。例えば、50回の繰り返し復号を行うためには、符号長である90個の符号(受信データ)を1フレームとして、その1フレームを受信する間に、 $538 \times 50 = 26900$  クロック動作することが必要であり、受信周波数の約300( $\div 26900/90$ )倍の高速動作が必要になる。受信周波数が数十MHzであるとする、GHz以上の速度での動作を要求されることになる。

20      また、図9の復号装置を、例えば、50台接続して、LDPC符号を復号する場合には、1フレーム(frame)目のバリエブルノード演算を行っている間に、2フレーム目のチェックノード演算を行い、3フレーム目のバリエブルノード演算を行う、というように、複数のバリエブルノード演算とチェックノード演算とを同時に行うことができる。この場合、90個の符号を受信する間に、269個の枝を計算すればよいので、復号装置は、受信周波数の約3( $\div 269/90$ )倍の周波数で動作すればよいことになり、十分に実現可能である。しかしながら、この場合、回路規模が、単純には、図9の復号装置の50倍になる。

25

次に、全ノードの演算を同時に行うことによって復号を行う場合(full parallel decoding)の復号装置の実装法について説明する。

この実装法については、例えば、C. Howland and A. Blanksby, "Parallel Decoding Architectures for Low Density Parity Check Codes", Symposium on Circuits and Systems, 2001 に記載されている。

図12A乃至図12Cは、図8の検査行列で表現される符号(符号化率2/3、符号長90)を復号する復号装置の一例の構成を示している。なお、図12Aは、復号装置全体の構成を示している。また、図12Bは、図12Aの復号装置の点線Bで囲まれた図中上部の詳細構成を示し、図12Cは、図12Aの復号装置の点線Cで囲まれた図中下部の詳細構成を示している。

図12A乃至図12Cの復号装置は、1つの受信用メモリ205、2つの枝入れ替え装置200および203、2つの枝用メモリ202および206、30個のチェックノード計算器201<sub>1</sub>乃至201<sub>30</sub>から構成されるチェックノード計算器201、90個のバリアブルノード計算器204<sub>1</sub>乃至204<sub>90</sub>から構成されるバリアブルノード計算器204からなる。

図12A乃至図12Cの復号装置では、枝用メモリ202または206から、269個ある枝に対応するメッセージデータを全て同時に読み出し、そのメッセージデータを用いて、269個の枝に対応する新たなメッセージデータを演算する。さらに、その演算の結果求められた新たなメッセージデータが全て同時に後段の枝用メモリ206または202に格納されていく。そして、図12A乃至図12Cの復号装置を繰り返し用いることで繰り返し復号が実現される。以下、各部について詳細に説明する。

枝用メモリ206は、前段のバリアブルノード計算器204<sub>1</sub>乃至204<sub>90</sub>からのメッセージD206<sub>1</sub>乃至D206<sub>90</sub>を全て同時に格納し、次の時刻(次のクロックのタイミング)に、メッセージD206<sub>1</sub>乃至D206<sub>90</sub>を、メッセージD207<sub>1</sub>乃至D207<sub>90</sub>として読み出し、次段の枝入れ替え装置200に、メッセージ200(D200<sub>1</sub>乃至D200<sub>90</sub>)として供給する。枝入れ替え装置200は、枝用メモリ206から

供給されたメッセージ  $D200_1$  乃至  $D200_{90}$  の順番を、図 8 の検査行列にしたがって並び替え（入れ替え）、チェックノード計算器  $201_1$  乃至  $201_{30}$  に、メッセージ  $D201_1$  乃至  $D201_{30}$  として供給する。

5      チェックノード計算器  $201_1$  乃至  $201_{30}$  は、枝入れ替え装置  $200$  から供給されるメッセージ  $D201_1$  乃至  $D201_{30}$  を用いて式 (7) にしたがって演算を行い、その演算の結果得られるメッセージ  $D202_1$  乃至  $D202_{30}$  を、枝用メモリ  $202$  に供給する。

10      枝用メモリ  $202$  は、前段のチェックノード計算器  $201_1$  乃至  $201_{30}$  から供給されるメッセージ  $D202_1$  乃至  $D202_{30}$  を全て同時に格納し、次の時刻に、そのすべてのメッセージ  $D202_1$  乃至  $D202_{30}$  を、メッセージ  $D203_1$  乃至  $D203_{30}$  として、次段の枝入れ替え装置  $203$  に供給する。

枝入れ替え装置  $203$  は、枝用メモリ  $202$  から供給されたメッセージ  $D203_1$  乃至  $D203_{30}$  の順番を図 8 の検査行列にしたがって並び替え、バリエブルノード計算器  $204_1$  乃至  $204_{90}$  に、メッセージ  $D204_1$  乃至  $D204_{90}$  として供給する。

15      バリエブルノード計算器  $204_1$  乃至  $204_{90}$  は、枝入れ替え装置  $203$  から供給されるメッセージ  $D204_1$  乃至  $D204_{90}$  と、受信用メモリ  $205$  から供給される受信データ  $D205_1$  乃至  $D205_{90}$  を用いて式 (1) にしたがって演算を行い、その演算の結果得られるメッセージ  $D206_1$  乃至  $D206_{90}$  を、次段の枝用メモリ  $206$  に供給する。

20      図 13 は、チェックノード演算を同時に行う図 12A 乃至図 12C のチェックノード計算器  $201_m$  ( $m=1, 2, \dots, 30$ ) の構成例を示している。

図 13 のチェックノード計算器  $201_m$  では、図 10 のチェックノード計算器  $101$  と同様にして、式 (7) のチェックノード演算が行われるが、そのチェックノード演算が、すべての枝について同時に行われる。

25      即ち、図 13 のチェックノード計算器  $201_m$  では、枝入れ替え装置  $200$  から供給される図 8 の検査行列の各列に対応するバリエブルノードからのメッセージ  $D221_1$  乃至  $D221_9$  ( $v_i$ ) が全て同時に読み込まれ、それぞれの下位 5 ビットで

ある絶対値  $D222_1$  乃至  $D222_9$  ( $|v_i|$ ) が LUT 2 2 1<sub>1</sub> 乃至 2 2 1<sub>9</sub> にそれぞれ供給される。また、メッセージ  $D221_1$  乃至  $D221_9$  ( $v_i$ ) の最上位ビットである 1 ビットの符号ビット  $D223_1$  乃至  $D223_9$  が、EXOR 回路 2 2 6<sub>1</sub> 乃至 2 2 6<sub>9</sub> にそれぞれ供給されるとともに、EXOR 回路 2 2 5 に供給される。

- 5     LUT 2 2 1<sub>1</sub> 乃至 2 2 1<sub>9</sub> は、絶対値  $D222_1$  乃至  $D222_9$  ( $|v_i|$ ) に対して、式 (7) における  $\phi(|v_i|)$  の演算を行った 5 ビットの演算結果  $D224_1$  乃至  $D224_9$  ( $\phi(|v_i|)$ ) をそれぞれ読み出し、それぞれを減算器 2 2 3<sub>1</sub> 乃至 2 2 3<sub>9</sub> に供給する。また、LUT 2 2 1<sub>1</sub> 乃至 2 2 1<sub>9</sub> は、演算結果  $D224_1$  乃至  $D224_9$  ( $\phi(|v_i|)$ ) を加算器 2 2 2 に供給する。
- 10    加算器 2 2 2 は、演算結果  $D224_1$  乃至  $D224_9$  ( $\phi(|v_i|)$ ) の値の総和 (1 行分の演算結果の総和) を演算し、9 ビットの演算結果  $D225$  ( $i = 1$  から 9 の  $\sum \phi(|v_i|)$ ) を、減算器 2 2 3<sub>1</sub> 乃至 2 2 3<sub>9</sub> に供給する。減算器 2 2 3<sub>1</sub> 乃至 2 2 3<sub>9</sub> は、演算結果  $D225$  から、演算結果  $D224_1$  乃至  $D224_9$  ( $\phi(|v_i|)$ ) をそれぞれ減算し、5 ビットの減算値  $D227_1$  乃至  $D227_9$  を、LUT 2 2 4<sub>1</sub> 乃至 2 2 4<sub>9</sub> に供給する。即ち、減算器 2 2 3<sub>1</sub> 乃至 2 2 3<sub>9</sub> は、全ての枝からのメッセージ  $v_i$  から求められた  $\phi(|v_i|)$  の積算値から、求めたい枝からのメッセージ  $v_i$  から求められた  $\phi(|v_i|)$  を減算して、その減算値  $D227_1$  乃至  $D227_9$  ( $i = 1$  から 8 までの  $\sum \phi(|v_i|)$ ) を LUT 2 2 4<sub>1</sub> 乃至 2 2 4<sub>9</sub> にそれぞれ供給する。LUT 2 2 4<sub>1</sub> 乃至 2 2 4<sub>9</sub> は、減算値  $D227_1$  乃至  $D227_9$  に対して、式 (7) における  $\phi^{-1}(\sum \phi(|v_i|))$  の演算を行った 5 ビットの演算結果  $D228_1$  乃至  $D228_9$  を読み出して出力する。
- 20

一方、EXOR 回路 2 2 5 は、全ての符号ビット  $D223_1$  乃至  $D223_9$  の排他的論理和を演算することにより、符号ビット  $D223_1$  乃至  $D223_9$  の乗算を行い、1 ビットの乗算値  $D226$  (1 行分の符号ビットの乗算値 ( $i = 1$  から 9 までの  $\prod \text{sign}(v_i)$ )) を EXOR 回路 2 2 6<sub>1</sub> 乃至 2 2 6<sub>9</sub> にそれぞれ供給する。EXOR 回路 2 2 6<sub>1</sub> 乃至 2 2 6<sub>9</sub> は、乗算値  $D226$  と符号ビット  $D223_1$  乃至  $D223_9$  それぞれとの排他的論理を演算することにより、乗算値  $D226$  を、符号ビット  $D223_1$  乃至  $D223_9$

- 25    ( $v_i$ ) を EXOR 回路 2 2 6<sub>1</sub> 乃至 2 2 6<sub>9</sub> にそれぞれ供給する。EXOR 回路 2 2 6<sub>1</sub> 乃至 2 2 6<sub>9</sub> は、乗算値  $D226$  と符号ビット  $D223_1$  乃至  $D223_9$  それぞれとの排他的論理を演算することにより、乗算値  $D226$  を、符号ビット  $D223_1$  乃至  $D223_9$

それぞれで除算した1ビットの除算値  $D229_i$  乃至  $D229_8$  ( $i = 1$  から 8 までの  $\Pi \text{sign}(v_i)$ ) を求めて出力する。

チェックノード計算器 201<sub>1</sub> では、LUT 224<sub>1</sub> 乃至 224<sub>8</sub> から出力された5ビットの演算結果  $D228_1$  乃至  $D228_8$  それぞれを下位5ビットとするとともに、

- 5 EXOR 回路 226<sub>1</sub> 乃至 226<sub>8</sub> から出力された除算値  $D229_1$  乃至  $D229_8$  それぞれを最上位ビットとする合計6ビットが、チェックノード演算の結果得られるメッセージ  $D230_1$  乃至  $D230_8$  として出力される。

以上のように、チェックノード計算器 201<sub>1</sub> では、式(7)の演算が行われ、メッセージ  $u_j$  が求められる。

- 10 なお、図13では、各メッセージが符号ビットを合わせて合計6ビットに量子化されているものとして、チェックノード計算器 201<sub>1</sub> を表している。また、図13の回路は1つのチェックノードに相当する。ここで処理の対象としている図8の検査行列については、その行数である30行のチェックノードが存在するため、図12A乃至図12Cの復号装置は、図13に示したようなチェックノード計算器 201<sub>1</sub> を30個有している。

- ここで、図13のチェックノード計算器 201<sub>1</sub> では、9個のメッセージを同時に計算することができる。そして、ここで処理の対象としている図8の検査行列の行の重みは、第1行が8で、第2乃至第30行が9であるため、即ち、チェックノードに供給されるメッセージの数が、8のケースが1つと、9のケースが29あるため、チェックノード計算器 201<sub>1</sub> は、図13の回路と同様の8つのメッセージを同時に計算することができる回路構成となっており、残りのチェックノード計算器 201<sub>2</sub> 乃至 201<sub>30</sub> は、図13の回路と同一構成となっている。

図14は、バリアブルノード演算を同時に行う図12A乃至図12Cのバリアブルノード計算器 204<sub>p</sub> ( $p = 1, 2, \dots, 90$ ) の構成例を示している。

- 25 図14のバリアブルノード計算器 204<sub>p</sub> では、図11のバリアブルノード計算器 103と同様にして、式(1)のバリアブルノード演算が行われるが、そのバリアブルノード演算が、すべての枝について同時に行われる。

即ち、図 1.4 のバリエブルノード計算器 204<sub>p</sub>では、枝入れ替え装置 203 から供給される、検査行列の各行に対応するチェックノードからの 6 ビットのメッセージ D251<sub>1</sub>乃至 D251<sub>5</sub> (メッセージ  $u_j$ ) が全て同時に読み込まれ、それぞれ加算器 252<sub>1</sub>乃至 252<sub>5</sub>に供給されるとともに、加算器 251に供給される。

- 5 また、バリエブルノード計算器 204<sub>p</sub>には、受信用メモリ 205から受信データ D271 が供給され、その受信データ D271 は、加減算器 252<sub>1</sub>乃至 252<sub>5</sub>に供給される。

- 加算器 251 は、全てのメッセージ D251<sub>1</sub>乃至 D251<sub>5</sub> (メッセージ  $u_j$ ) を積算し、9 ビットの積算値 D252 (1 列分のメッセージの総和値 ( $j = 1$  から 5 までの  $\sum u_j$ )) を加減算器 252<sub>1</sub>乃至 252<sub>5</sub>に供給する。加減算器 252<sub>1</sub>乃至 252<sub>5</sub>は、加算値 D252 から、メッセージ D251<sub>1</sub>乃至 D251<sub>5</sub> (メッセージ  $u_j$ ) をそれぞれ減算する。即ち、加減算器 252<sub>1</sub>乃至 252<sub>5</sub>は、全ての枝からのメッセージ  $u_j$  の積算値 D252 から、求めたい枝からのメッセージ D251<sub>1</sub>乃至 D251<sub>5</sub> (メッセージ  $u_j$ ) をそれぞれ減算して、その減算値 ( $j = 1$  から 4 までの  $\sum u_j$ ) を
- 15 求める。

さらに、加減算器 252<sub>1</sub>乃至 252<sub>5</sub>は、減算値 ( $j = 1$  から 4 までの  $\sum u_j$ ) に、受信データ D271 ( $u_{0i}$ ) を加算して、6 ビットの加算値 D253<sub>1</sub>乃至 D253<sub>5</sub>を、バリエブルノード演算の結果として出力する。

- 以上のように、バリエブルノード計算器 204<sub>p</sub>では、式 (1) の演算が行われ、メッセージ  $v_i$  が求められる。
- 20

- なお、図 1 4 では、各メッセージが符号ビットを合わせて合計 6 ビットに量子化されているものとして、バリエブルノード計算器 204<sub>p</sub>を表している。また、図 1 4 の回路は 1 つのバリエブルノードに相当する。ここで処理の対象としている図 8 の検査行列については、その列数である 90 列のバリエブルノードが存在するから、図 1 2 A 乃至図 1 2 C の復号装置は、図 1 4 に示したような回路を 90 個有している。
- 25

ここで、図 1 4 のバリエブルノード計算器 204<sub>p</sub>では、5 個のメッセージを

同時に計算することができる。そして、ここで処理の対象としている図 8 の検査行列は、重みが 5, 3, 2, 1 の列が、それぞれ、15 列、45 列、29 列、1 列あるので、バリアブルノード計算器 204<sub>1</sub>乃至 204<sub>90</sub>のうちの 15 個は、図 14 の回路と同一構成となっており、残りの 45 個、29 個、1 個は、図 14 の回路と同様の 3, 2, 1 つのメッセージをそれぞれ同時に計算することができる回路構成となっている。

なお、図示しないが、図 12A 乃至図 12C の復号装置においても、図 9 における場合と同様に、復号の最終段においては、式 (1) のバリアブルノード演算の代わりに、式 (5) の演算が行われ、その演算結果が最終的な復号結果として出力される。

図 12A 乃至図 12C の復号装置によれば、269 個ある枝に対応するメッセージすべてを 1 クロックで同時に計算することができる。

図 12A 乃至図 12C の復号装置を繰り返し用いて復号する場合には、チェックノード演算とバリアブルノード演算とを交互に行い、1 回の復号を 2 クロックで行うことができる。従って、例えば、50 回の復号を行うためには、符号長が 90 個の符号を 1 フレームとする受信データを受信する間に  $2 \times 50 = 100$  クロック動作すれば良いことになり、ほぼ受信周波数と同一の動作周波数でよいことになる。一般的に、LDPC 符号は、符号長が数千から数万と大きいことから、図 12A 乃至図 12C の復号装置を用いれば、復号回数を極めて多くすることができ、誤り訂正性能の向上を期待することができる。

しかしながら、図 12A 乃至図 12C の復号装置は、タナグラフのすべての枝に対応するメッセージの演算を、並列で行うため、回路規模が、符号長に比例して大きくなる。また、図 12A 乃至図 12C の復号装置を、ある符号長の、ある符号化率の、ある検査行列を持つ LDPC 符号の復号を行う装置として構成した場合、その復号装置において、他の符号長や、他の符号化率、他の検査行列を持つ LDPC 符号の復号を行うことは困難となる。即ち、図 12A 乃至図 12C の復号装置は、図 9 の復号装置のように、制御信号を変えるだけでは、様々な符号を

復号することに対処することが困難であり、符号依存性が高い。

図 9 および図 12 A 乃至図 12 C の復号装置の他に、一つでも全てでもなく、4 つずつのメッセージの計算を同時に行う実装法について、例えば、E. Yeo, P. Pakzad, B. Nikolic and V. Anantharam, "VLSI Architectures for iterative Decoders in Magnetic Recording Channels", IEEE Transactions on Magnetics, Vol. 37, No. 2, March 2001 に述べられているが、この場合、メモリの異なるアドレスからの同時読み出し、もしくは同時書き込みを避けることが一般的には容易でなく、メモリアクセス制御が困難であるという問題がある。

- 10     また、サンプロダクトアルゴリズムを近似して実装する方法なども提案されているが、この方法では、性能の劣化を招いてしまう。 サンプロダクトアルゴリズムをハードウェアに実装する場合には、上述したように、枝に対応するメッセージの演算（チェックノード演算とビットノード(bit node)計算）を、1 つずつシリアル(serial)に行う方法、すべて並列（フルパラレル(full parallel)）に行う方法、幾つかずつ並列（パラレル(parallel)）に行う方法がある。

- 20     しかしながら、枝に対応するメッセージの演算を 1 つずつ行う方法では、高い動作周波数が必要となる。そこで、スループット(throughput)を上げる方法として、装置を、パイプライン(pipeline)化する方法があるが、この場合、回路規模、特にメモリ（の容量）が大きくなってしまう。

また、メッセージの演算を全て並列に行う方法では、ロジック(logic)の回路規模が大きくなるとともに、符号依存性が高くなる。

さらに、メッセージの演算を、幾つかずつ並列に行う方法では、メモリアクセスの制御が難しくなる。

25

#### 発明の開示

本発明は、このような状況に鑑みてなされたものであり、ロジック、メモリ共



に回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことができるようにするものである。

本発明の復号装置は、 $P \times P$  の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または  $P \times P$  の 0 行列を構成行列として、LDPC 符号の検査行列が、複数の構成行列の組合せで表される場合において、LDPC 符号の復号のための  $P$  個のチェックノードの演算を同時に行う第 1 の演算手段と、LDPC 符号の復号のための  $P$  個のバリエブルノードの演算を同時に行う第 2 の演算手段とを備えることを特徴とする。

第 1 の演算手段は、チェックノードの演算を行う  $P$  個のチェックノード計算器を有し、第 2 の演算手段は、バリエブルノードの演算を行う  $P$  個のバリエブルノード計算器を有するようにすることができる。

$P$  個のチェックノードの演算、または  $P$  個のバリエブルノードの演算の結果得られる  $P$  個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段をさらに備えるようにすることができる。

メッセージ記憶手段は、チェックノードの演算時に読み出される枝に対応するメッセージデータを、検査行列の 1 を行方向に詰めるように格納するようにすることができる。

メッセージ記憶手段は、バリエブルノード演算時に読み出される枝に対応するメッセージデータを、検査行列の 1 を列方向に詰めるように格納するようにすることができる。

メッセージ記憶手段は、検査行列を表す構成行列のうちの、重みが 2 以上の構成行列について、その構成行列を、重みが 1 の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが 1 の単位行列、準単位行列、またはシフト行列に属する  $P$  個の枝に対応するメッセージを、同一のアドレスに格納するようにすることができる。

メッセージ記憶手段は、行数／P 個の FIFO と、列数／P 個の FIFO とで構成され、行数／P 個の FIFO と列数／P 個の FIFO は、それぞれ、検査行列の行と列の重みに対応するワード数を有するようにすることができる。

- 5      メッセージ記憶手段は、RAM(Random Access Memory)で構成され、RAM は、メッセージデータを、読み出される順番に詰めて格納し、格納位置順に読み出すようにすることができる。

LDPC 符号の受信情報を格納するとともに、P 個の受信情報を同時に読み出す受信情報記憶手段をさらに備えるようにすることができる。

- 10     受信情報記憶手段は、受信情報を、バリエブルノードの演算に必要となる順番に読み出すことができるように格納するようにすることができる。

P 個のチェックノードの演算、または P 個のバリエブルノードの演算の結果得られるメッセージを並べ替える並べ替え手段をさらに備えるようにすることができる。

並べ替え手段は、バレルシフタで構成されるようにすることができる。

- 15     第 1 の演算手段と第 2 の演算手段は、P 個の枝に対応するメッセージを求めるようにすることができる。

第 1 の演算手段は、P 個のチェックノードの演算と P 個のバリエブルノードの演算の一部とを行い、第 2 の演算手段は、P 個のバリエブルノードの演算の他の一部を行うようにすることができる。

- 20     第 1 の演算手段は、P 個のチェックノードの演算と P 個のバリエブルノードの演算の一部を行う P 個の計算器を有し、第 2 の演算手段は、P 個のバリエブルノードの演算の他の一部を行う P 個の計算器を有するようにすることができる。

- 25     第 1 の演算手段が P 個のチェックノードの演算と P 個のバリエブルノードの演算の一部を行うことにより得られる P 個の枝に対応する第 1 の復号途中結果を同時に読み書きする第 1 の復号途中結果記憶手段をさらに備えるようにすることができる。

第 1 の復号途中記憶手段は、P 個のバリエブルノードの演算の他の一部を行う

時に読み出される枝に対応する第1の復号途中結果を、検査行列の1を行方向に詰めるように格納するようにすることができる。

第1の復号途中結果記憶手段は、2個のシングルポート RAM(Random Access Memory)であるようにすることができる。

- 5      2個のシングルポート RAM は、第1の復号途中結果をP個ずつ交互に格納するようにすることができる。

2個のシングルポート RAM(Random Access Memory)は、それぞれ同一のアドレスに格納している第1の復号途中結果を読み出すようにすることができる。

- 10      第1の復号途中結果記憶手段は、検査行列を表す構成行列のうちの、重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列に属するP個の枝に対応する第1の復号途中結果を、同一のアドレスに格納するようにすることができる。

- 15      第2の演算手段がP個のバリエブルノードの演算の他の一部を行うことにより得られるP個の枝に対応する第2の復号途中結果を同時に読み書きする第2の復号途中結果記憶手段をさらに備えるようにすることができる。

LDPC 符号の受信情報を格納するとともに、P個の受信情報を同時に読み出す受信情報記憶手段をさらに備えるようにすることができる。

- 20      受信情報記憶手段は、受信情報を、P個のバリエブルノードの演算の他の一部の演算に必要となる順番に読み出すことができるように格納するようにすることができる。

- 25      第1の演算手段がP個のチェックノードの演算とP個のバリエブルノードの演算の一部を行うことにより得られる第1の復号途中結果、または第2の演算手段がP個のバリエブルノードの演算の他の一部を行うことにより得られる第2の復号途中結果を並べ替える並べ替え手段をさらに備えるようにすることができる。

並べ替え手段は、バレルシフタで構成されるようにすることができる。

第1の演算手段は、P個のチェックノードの演算の一部を行い、第2の演算手段は、P個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算とを行うようにすることができる。

- 5 第1の演算手段は、P個のチェックノードの演算の一部を行うP個の計算器を有し、第2の演算手段は、P個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算を行うP個の計算器を有することができる。

第1の演算手段がP個のチェックノードの演算の一部を行うことにより得られるP個の枝に対応する第1の復号途中結果を同時に読み書きする第1の復号途中結果記憶手段をさらに備えるようにすることができる。

- 10 第2の演算手段がP個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算を行うことにより得られるP個の枝に対応する第2の復号途中結果を同時に読み書きする第2の復号途中結果記憶手段をさらに備えるようにすることができる。

- 15 第2の復号途中結果記憶手段は、P個のチェックノードの演算の他の一部と、P個のバリエブルノードの演算を行う時に読み出される枝に対応する第2の復号途中結果を、検査行列の1を列方向に詰めるように格納するようにすることができる。

第2の復号途中結果記憶手段は、2個のシングルポートRAM(Random Access Memory)であるようにすることができる。

- 20 2個のシングルポートRAMは、第2の復号途中結果をP個ずつ交互に格納するようにすることができる。

2個のシングルポートRAM(Random Access Memory)は、それぞれ同一のアドレスに格納している第2の復号途中結果を読み出すようにすることができる。

- 25 第2の復号途中結果記憶手段は、検査行列を表す構成行列のうちの、重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列に属するP個の枝に対応する第2の復号途中結果を、

同一のアドレスに格納するようにすることができる。

LDPC 符号の受信情報を格納するとともに、P 個の受信情報を同時に読み出す受信情報記憶手段をさらに備えるようにすることができる。

請求の範囲第 3 6 項に記載の復号装置であって、受信情報記憶手段は、受信情報  
5 報を、P 個のチェックノードの演算の他の一部と、P 個のバリアブルノードの演算に必要となる順番に読み出すことができるように格納するようにすることができる。

第 1 の演算手段が P 個のチェックノードの演算の一部を行うことにより得られる第 1 の復号途中結果、または第 2 の演算が P 個のチェックノードの演算の  
10 他の一部と、P 個のバリアブルノードの演算を行うことにより得られる第 2 の復号途中結果を並べ替える並べ替え手段をさらに備えるようにすることができる。

並べ替え手段は、バレルシフトで構成されるようにすることができる。

本発明の復号方法は、 $P \times P$  の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、単位行列もしくは準  
15 単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または  $P \times P$  の 0 行列を構成行列として、LDPC 符号の検査行列が、複数の構成行列の組合せで表される場合において、LDPC 符号の復号のための P 個のチェックノードの演算を同時に行う第 1 の演算ステップと、LDPC 符号の復号のための P 個のバリアブルノードの演算を同時に行う第 2 の演算ステップとを含むことを特徴とする。  
20

本発明のプログラムは、LDPC 符号の復号のための P 個のチェックノードの演算を同時に行う第 1 の演算ステップと、LDPC 符号の復号のための P 個のバリアブルノードの演算を同時に行う第 2 の演算ステップとを含むことを特徴とする。

本発明においては、 $P \times P$  の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、  
25 もしくはシフト行列のうちの複数の和である和行列、または  $P \times P$  の 0 行列を構

成行列として、LDPC 符号の検査行列が、複数の構成行列の組合せで表される場合において、LDPC 符号の復号のための  $P$  個のチェックノードの演算が同時に行われ、LDPC 符号の復号のための  $P$  個のバリエブルノードの演算が同時に行われる。

5

#### 図面の簡単な説明

図 1 は、LDPC 符号の検査行列  $H$  を説明する図である。

図 2 は、LDPC 符号の復号手順を説明するフローチャートである。

図 3 は、メッセージの流れを説明する図である。

10 図 4 は、LDPC 符号の検査行列の例を示す図である。

図 5 は、検査行列のタナグラフを示す図である。

図 6 は、バリエブルノードを示す図である。

図 7 は、チェックノードを示す図である。

図 8 は、LDPC 符号の検査行列の例を示す図である。

15 図 9 は、ノード演算を一つずつ行う LDPC 符号の復号装置の構成例を示すブロック図である。

図 10 は、メッセージを一つずつ計算するチェックノード計算器の構成例を示すブロック図である。

20 図 11 は、メッセージを一つずつ計算するバリエブルノード計算器の構成例を示すブロック図である。

図 12 A は、ノード演算を全て同時に行う LDPC 符号の復号装置の構成例を示すブロック図である。

図 12 B は、ノード演算を全て同時に行う LDPC 符号の復号装置の構成例を示すブロック図である。

25 図 12 C は、ノード演算を全て同時に行う LDPC 符号の復号装置の構成例を示すブロック図である。

図 13 は、メッセージを同時に計算するチェックノード計算器の構成例を示す

ブロック図である。

図 1 4 は、メッセージを同時に計算するバリアブルノード計算器の構成例を示すブロック図である。

図 1 5 は、 $5 \times 5$  単位に分割した検査行列を示す図である。

5 図 1 6 A は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

図 1 6 B は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

10 図 1 6 C は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

図 1 7 は、図 1 6 A 乃至図 1 6 C の復号装置の復号処理を説明するフローチャートである。

図 1 8 は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

15 図 1 9 は、チェックノード計算器の構成例を示すブロック図である。

図 2 0 は、バリアブルノード計算器の構成例を示すブロック図である。

図 2 1 は、図 1 8 の計算器の構成例を示すブロック図である。

図 2 2 は、図 1 8 の計算器の構成例を示すブロック図である。

20 図 2 3 は、図 1 8 の復号途中結果格納用メモリの構成例を示すブロック図である。

図 2 4 は、図 1 8 の復号途中結果格納用 RAM の動作を説明するタイミングチャートである。

図 2 5 は、図 1 8 の復号装置の復号処理を説明するフローチャートである。

25 図 2 6 は、本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

図 2 7 は、チェックノード計算器の構成例を示すブロック図である。

図 2 8 は、バリアブルノード計算器の構成例を示すブロック図である。

図 29 は、図 26 の計算器の構成例を示すブロック図である。

図 30 は、図 26 の計算器の構成例を示すブロック図である。

図 31 は、図 26 の復号途中結果格納用メモリの構成例を示すブロック図である。

- 5 図 32 は、図 31 の復号途中結果格納用 RAM の動作を説明するタイミングチャートである。

図 33 は、図 26 の復号装置の復号処理を説明するフローチャートである。

図 34 は、本発明を適用したコンピュータの一実施の形態の構成例を示すブロック図である。

10

発明を実施するための最良の形態

以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳細に説明する。

- 15 図 15 は、 $5 \times 5$  の行列の単位に間隔を空けた  $30 \times 90$  の検査行列の例を示している。なお、図 15 の検査行列自体は、図 8 に示した検査行列と同一である。

- 図 15 においては、検査行列は、 $5 \times 5$  の単位行列、その単位行列の 1 のうち 1 個以上が 0 になった行列（以下、適宜、準単位行列という）、単位行列または準単位行列をサイクリックシフト(cyclic shift)した行列（以下、適宜、シフト行列という）、単位行列、準単位行列、またはシフト行列のうちの 2 以上(複  
20 数)の和（以下、適宜、和行列という）、 $5 \times 5$  の 0 行列の組合わせで表わされている。なお、図 15 の検査行列で表現される LDPC 符号は、符号化率  $2/3$ 、符号長 90 である。

- 図 15 の検査行列は、 $5 \times 5$  の単位行列、準単位行列、シフト行列、和行列、0 行列で構成されているといえることができる。そこで、検査行列を構成する、こ  
25 れらの  $5 \times 5$  の行列を、以下、適宜、構成行列という。

図 16 A 乃至図 16 C は、図 15 の検査行列で表現される LDPC 符号を復号する復号装置の一実施の形態の構成例を示している。なお、図 16 A は、復号装置



の全体の構成を示している。また、図 1 6 B は、図 1 6 A の復号装置の点線 B で囲まれた図中左部の詳細構成を示し、図 1 6 C は、図 1 6 A の復号装置の点線 C で囲まれた図中右部の詳細構成を示している。

図 1 6 A 乃至図 1 6 C の復号装置 3 0 0 は、スイッチ 3 1 0 および 3 1 5、6  
5 つの FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> からなる枝データ格納メモリ 3 1 1、セクタ 3  
1 2、5 つのチェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> からなるチェックノード  
計算部 3 1 3、2 つのサイクリックシフト回路 3 1 4 および 3 2 0、1 8 個の  
FIFO 3 1 6<sub>1</sub> 乃至 3 1 6<sub>18</sub> からなる枝データ格納メモリ 3 1 6、セクタ 3 1 7、  
受信情報を格納する受信データ用メモリ 3 1 8、パリアブルノード計算部 3 1 9、  
10 制御部 3 2 1 から構成される。

この復号装置 3 0 0 の各部について詳細に説明する前に、まず、枝データ格納メモリ 3 1 1 と 3 1 6 へのデータの格納方法について説明する。

枝データ格納メモリ 3 1 1 は、検査行列の行数 3 0 を構成行列の行数 5 で除算した数である 6 つの FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> から構成されている。FIFO 3 1 1<sub>y</sub>  
15 ( $y = 1, 2, \dots, 6$ ) は、構成行列の行数および列数である 5 つの枝に対応するメッセージを同時に読み出しもしくは書き込むことができるようになっており、その長さ（段数）は、検査行列の行方向の 1 の数（ハミング重み）の最大数である 9 になっている。

FIFO 3 1 1<sub>1</sub> には、図 1 5 の検査行列の第 1 行目から第 5 行目までの 1 の位置  
20 に対応するデータが、各行共に横方向（列方向）に詰めた形に（0 を無視した形で）格納される。すなわち、第  $j$  行第  $i$  列を、 $(j, i)$  と表すこととすると、FIFO 3 1 1<sub>1</sub> の第 1 の要素（第 1 段）には、検査行列の  $(1, 1)$  から  $(5, 5)$  の  $5 \times 5$  の単位行列の 1 の位置に対応するデータが格納される。第 2 の要素には、検査行列の構成行列である  $(1, 21)$  から  $(5, 25)$  のシフト行列（ $5 \times 5$  の単位行列を右方向に  
25 3 つだけサイクリックシフトしたシフト行列）の 1 の位置に対応するデータが格納される。第 3 から第 8 の要素も同様に検査行列の構成行列と対応づけてデータが格納される。そして、第 9 の要素には、検査行列の  $(1, 86)$  から  $(5, 90)$  のシフ

ト行列（ $5 \times 5$ の単位行列のうちの1行目の1を0に置き換えて1つだけ左にサイクリックシフトしたシフト行列）の1の位置に対応するデータが格納される。ここで、検査行列の(1, 86)から(5, 90)のシフト行列においては、1行目に1がないため、FIFO 3 1 1<sub>1</sub>の1行目のみ要素数は8、残りの行は要素数が9となる。

- 5 FIFO 3 1 1<sub>2</sub>には、図15の検査行列の第6行目から第10行目までの1の位置に対応するデータが格納される。すなわち、FIFO 3 1 1<sub>2</sub>の第1の要素には、検査行列の(6, 1)から(10, 5)の和行列（ $5 \times 5$ の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2のシフト行列の和である和行列）を構成する第1のシフト行列の1の位置に対応するデータが格納される。また、第2の要素には、検査行列の(6, 1)から(10, 5)の和行列を構成する第2のシフト行列の1の位置に対応するデータが格納される。
- 10

即ち、重みが2以上の構成行列については、その構成行列を、重みが1である $P \times P$ の単位行列、そのコンポーネントである1のうち1個以上が0になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列の1の位置に対応するデータ（単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ）は、同一アドレス（FIFO 3 1 1<sub>1</sub>乃至3 1 1<sub>6</sub>のうちの同一のFIFO）に格納される。

15

- 20 以下、第3から第9の要素についても、検査行列に対応づけてデータが格納される。FIFO 3 1 1<sub>2</sub>は全行共に要素数は9となる。

FIFO 3 1 1<sub>3</sub>乃至3 1 1<sub>6</sub>も同様に検査行列に対応づけてデータを格納し、各FIFO 3 1 1<sub>3</sub>乃至3 1 1<sub>6</sub>それぞれの長さは9である。

- 枝データ格納メモリ 3 1 6は、検査行列の列数90を、構成行列の列数である5で割った18個のFIFO 3 1 6<sub>1</sub>乃至3 1 6<sub>18</sub>から構成されている。FIFO 3 1 6<sub>x</sub>（ $x = 1, 2, \dots, 18$ ）は、構成行列の行数および列数である5つの枝に対応するメッセージを同時に読み出しもしくは書き込むことができるようにな
- 25

っている。

FIFO 3 1 6<sub>1</sub>には、図 1 5 の検査行列の第 1 列目から第 5 列目までの 1 の位置に対応するデータが、各列共に縦方向（行方向）に詰めた形に（0 を無視した形で）格納される。すなわち、FIFO 3 1 6<sub>1</sub>の第 1 の要素（第 1 段）には、検査行列の (1, 1) から (5, 5) の 5 × 5 の単位行列の 1 の位置に対応するデータが格納される。第 2 の要素には、検査行列の (6, 1) から (10, 5) の和行列（5 × 5 の単位行列を右に 1 つだけサイクリックシフトした第 1 のシフト行列と、右に 2 つだけサイクリックシフトした第 2 のシフト行列との和である和行列）を構成する第 1 のシフト行列の 1 の位置に対応するデータが格納される。また、第 3 の要素には、検査行列の (6, 1) から (10, 5) の和行列を構成する第 2 のシフト行列の 1 の位置に対応するデータが格納される。

即ち、重みが 2 以上の構成行列については、その構成行列を、重みが 1 である P × P の単位行列、そのコンポーネントである 1 のうち 1 個以上が 0 になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが 1 の単位行列、準単位行列、またはシフト行列の 1 の位置に対応するデータ（単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ）は、同一アドレス（FIFO 3 1 6<sub>1</sub>乃至 3 1 6<sub>18</sub>のうちの同一の FIFO）に格納される。

以下、第 4 および第 5 の要素についても、検査行列に対応づけて、データが格納される。この FIFO 3 1 6<sub>1</sub>の要素数（段数）は、検査行列の第 1 列から第 5 列における行方向の 1 の数（ハミング重み）の最大数である 5 になっている。

FIFO 3 1 6<sub>2</sub>と 3 1 6<sub>3</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さ（段数）は、5 である。FIFO 3 1 6<sub>4</sub>乃至 3 1 6<sub>12</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さは 3 である。FIFO 3 1 6<sub>13</sub>乃至 3 1 6<sub>18</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さは 2 である。但し、FIFO 3 1 6<sub>18</sub>の第 1 の要素は、検査行列の (1, 86) から (5, 90) に相当し、第 5 列目（検査行列の (1, 90) から (5, 90)）に 1 がいないため、データは格納

されない。

以下、図 1 6 A 乃至図 1 6 C の復号装置 3 0 0 の各部の動作について詳細に説明する。 スイッチ 3 1 0 には、サイクリックシフト回路 3 2 0 から 5 つのメッセージ（データ） D319 が供給されるとともに、制御部 3 2 1 から検査行列のどの行に属するかの情報（Matrix データ）を表す制御信号 D320 が供給される。制御信号 D320 にしたがって、5 つのメッセージ（データ） D319 を格納する FIFO を、FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> の中から選択し、選択した FIFO に 5 つのメッセージデータ D319 をまとめて順番に供給していく。

枝データ格納メモリ 3 1 1 は、6 つの FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> からなる。枝データ格納メモリ 3 1 1 の FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> には、スイッチ 3 1 0 から、5 つのメッセージ D319 がまとめて順番に供給され、FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> は、5 つのメッセージ D319 をまとめて順番に（同時に）格納していく。また、枝データ格納メモリ 3 1 1 は、データを読み出す際には、FIFO 3 1 1<sub>1</sub> から 5 つのメッセージ（データ） D311<sub>1</sub> を順番に読み出し、次段のセクタ 3 1 2 に供給する。

枝データ格納メモリ 3 1 1 は、FIFO 3 1 1<sub>1</sub> からのメッセージ D311<sub>1</sub> の読み出しの終了後、FIFO 3 1 1<sub>2</sub> 乃至 3 1 1<sub>6</sub> からも、順番に、メッセージ D311<sub>1</sub> 乃至 D311<sub>6</sub> をそれぞれ読み出し、セクタ 3 1 2 に供給する。

セクタ 3 1 2 には、制御部 3 2 1 から、FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> のうち、メッセージデータを読み出す FIFO（現在データが読み出されている FIFO）の選択を表す選択信号 D321 が供給されるとともに、枝データ格納メモリ 3 1 1 から 5 つのメッセージ（データ） D311<sub>1</sub> 乃至 D311<sub>6</sub> が供給される。セクタ 3 1 2 は、選択信号 D321 にしたがって、FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>6</sub> のうちの、現在データが読み出されている FIFO を選択し、その選択した FIFO から供給された 5 つのメッセージデータを、メッセージ D312 として、チェックノード計算部 3 1 3 に供給する。

チェックノード計算部 3 1 3 は、5 つのチェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> からなる。チェックノード計算部 3 1 3 には、セクタ 3 1 2 を介して 5

つのメッセージ D312 が供給され、そのメッセージ D312 が、チェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> のそれぞれに 1 つずつ供給される。また、チェックノード計算部 3 1 3 には、制御部 3 2 1 から制御信号 D322 が供給され、その制御信号 D322 が、チェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> に供給される。チェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> は、メッセージ D312 を用いて、式 (7) にしたがって同時に演算を行い、その演算の結果、5 個の枝に対応するメッセージ D313 を求める。チェックノード計算部 3 1 3 は、チェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> による演算の結果得られる 5 つのメッセージ D313 をサイクリックシフト回路 3 1 4 に供給する。

- 10      ここで、制御部 3 2 1 からチェックノード計算部 3 1 3 に供給される制御信号 D322 は、図 10 の制御信号 D106 に対応するものであり、チェックノード計算部 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> それぞれは、図 10 に示したチェックノード計算器 1 0 1 と同様に構成される。

- 15      サイクリックシフト回路 3 1 4 には、チェックノード計算部 3 1 3 で計算された 5 つのメッセージ D313 が供給されるとともに、制御部 3 2 1 から、そのメッセージ D313 に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D323 が供給される。サイクリックシフト回路 3 1 4 は、制御信号 D323 を元に、5 つのメッセージ D313 を並べ替えるサイクリックシフトを行い、その結果をメッセージ D314 として、スイッチ 3 1 5 に供給する。

- 20      スイッチ 3 1 5 には、サイクリックシフト回路 3 1 4 から供給される 5 つのメッセージ (データ) D314 が検査行列のどの列に属するかの情報を表す制御信号 D324 が供給されるとともに、サイクリックシフト回路 3 1 4 から、メッセージ D314 が供給される。スイッチ 3 1 5 は、制御信号 D324 にしたがって、メッセージ D314 を格納する FIFO を、FIFO 3 1 6<sub>1</sub> 乃至 3 1 6<sub>18</sub> の中から選択し、選択した FIFO に 5 つのメッセージ D314 をまとめて順番に供給していく。

枝データ格納メモリ 3 1 6 は、18 個の FIFO 3 1 6<sub>1</sub> 乃至 3 1 6<sub>18</sub> からなる。

枝データ格納メモリ 316 の FIFO 316<sub>1</sub> 乃至 316<sub>18</sub> には、スイッチ 315 から 5 つのメッセージ D314 がまとめて順番に（同時に）供給され、FIFO 316<sub>1</sub> 乃至 316<sub>18</sub> は、その 5 つのメッセージ D314 をまとめて順番に格納していく。

- 5 また、枝データ格納メモリ 316 は、データを読み出す際には、FIFO 316<sub>1</sub> から 5 つのメッセージ D315<sub>1</sub> を順番に読み出し、次段のセクタ 317 に供給する。枝データ格納メモリ 316 は、FIFO 316<sub>1</sub> からのデータの読み出しの終了後、FIFO 316<sub>2</sub> 乃至 316<sub>18</sub> からも、順番に、メッセージ D315<sub>2</sub> 乃至 D31318 を読み出し、セクタ 317 に供給する。

- 10 セクタ 317 には、制御部 321 から FIFO 316<sub>1</sub> 乃至 316<sub>18</sub> のうち、メッセージデータを読み出す FIFO（現在データが読み出されている FIFO）の選択を表す選択信号 D325 が供給されるとともに、枝データ格納メモリ 316 からメッセージデータ D315<sub>1</sub> 乃至 D31318 が供給される。セクタ 317 は、選択信号 D325 にしたがって、FIFO 316<sub>1</sub> 乃至 316<sub>18</sub> のうちの、現在データが読み出されている FIFO を選択し、その選択した FIFO から供給される 5 つのメッセージデータを、メッセージ D316 として、バリアブルノード計算部 319 と上述した式（5）の演算を行う不図示のブロックに供給する。

- 一方、受信データ用メモリ 318 は、通信路を通して受信した受信信号から、受信 LLR（対数尤度比）を計算しており、その計算した受信 LLR を 5 つまとめて（同時に）受信データ D317（LDPC 符号）としてバリアブルノード計算部 319 と、式（5）の演算を行う不図示のブロックに供給する。なお、受信データ用メモリ 318 は、バリアブルノード計算部 319 のバリアブルノード演算に必要な順番に、受信データ D317 を読み出す。

- 25 バリアブルノード計算部 319 は、5 つのバリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> からなる。バリアブルノード計算部 319 には、セクタ 317 を介して 5 つのメッセージ D316 が供給され、そのメッセージ D316 が、バリアブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> のそれぞれに 1 つずつ供給される。また、バリアブルノード計算部 319 には、受信データ用メモリ 318 から 5 つの受信データ

タ D317 が供給され、その受信データ D317 が、バリエブルノード計算器 3 1 9<sub>1</sub> 乃至 3 1 9<sub>5</sub> のそれぞれに 1 つずつ供給される。さらに、バリエブルノード計算部 3 1 9 には、制御部 3 2 1 から制御信号 D326 が供給され、その制御信号 D326 がバリエブルノード計算器 3 1 9<sub>1</sub> 乃至 3 1 9<sub>5</sub> に供給される。

5     バリエブルノード計算器 3 1 9<sub>1</sub> 乃至 3 1 9<sub>5</sub> は、メッセージ D316 と、受信データ D317 を用いて、式 (1) にしたがって同時に演算を行い、その演算の結果、5 個の枝に対応するメッセージ D318 を求める。バリエブルノード計算部 3 1 9 は、バリエブルノード計算器 3 1 9<sub>1</sub> 乃至 3 1 9<sub>5</sub> の結果得られる 5 つのメッセージ D318 を、サイクリックシフト回路 3 2 0 に供給する。

10     ここで、制御部 5 2 1 からバリエブルノード計算部 3 1 9 に供給される制御信号 D326 は、図 1 1 の制御信号 D107 に対応するものであり、バリエブルノード計算器 3 1 9<sub>1</sub> 乃至 3 1 9<sub>5</sub> それぞれは、図 1 1 のバリエブルノード計算器 1 0 3 と同様に構成される。

サイクリックシフト回路 3 2 0 には、バリエブルノード計算部 3 1 9 から 5 つ  
15     のメッセージ D318 が供給されるとともに、制御部 3 2 1 から、そのメッセージ D318 に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報を表す制御信号 D327 が供給される。サイクリックシフト回路 3 2 0 は、制御信号 D327 を元に、メッセージ D327 を並べ替えるサイクリックシフトを行い、その結果をメッセージ D319 として、スイッチ 3  
20     1 0 に供給する。

なお、制御部 3 2 1 は、制御信号 D320 をスイッチ 3 1 0 に、選択信号 D321 をセレクタ 3 1 2 に供給することにより、それぞれを制御する。また制御部 3 2 1 は、制御信号 D322 をチェックノード計算部 3 1 3 に、制御信号 D323 をサイ  
25     クリックシフト回路 3 1 4 に、制御信号 D324 をスイッチ 3 1 5 に供給することにより、それぞれを制御する。さらに、制御部 3 2 1 は、選択信号 D325 をセレクタ 3 1 7、制御信号 D326 をバリエブルノード計算部 3 1 9 に、制御信号 D327 をサイクリックシフト回路 3 2 0 に供給することにより、それぞれを制御する。

以上の動作を1巡することで、LDPC符号の1回の復号を行うことができる。

図16A乃至図16Cの復号装置300は、所定の回数だけLDPC符号を復号した後、図示しないが、式(5)にしたがって最終的な復号結果を求めて出力する。

なお、枝データ(枝に対応するメッセージ)が欠けている箇所に関しては、メモリ格納時(枝データ格納メモリ311と316へのデータ格納時)には、何のメッセージも格納せず、また、ノード演算時(チェックノード計算部313でのチェックノード演算時とバリアブルノード計算部319でのバリアブルノード演算時)にも何の演算も行わない。

図17は、図16A乃至図16Cの復号装置300の復号処理を説明するフローチャートである。この処理は、例えば、受信データ用メモリ318に復号すべき受信データが格納されたとき、開始される。

ステップS31において、バリアブルノード計算部319は、バリアブルノード演算を行う。

具体的には、バリアブルノード計算部319には、セレクタ317を介して、5つのメッセージD316(メッセージ $u_j$ )が供給される。即ち、枝データ格納メモリ316は、後述するステップS39で格納されたFIFO316<sub>1</sub>から5つのメッセージD316<sub>1</sub>を順番に読み出し、その後、FIFO316<sub>2</sub>乃至316<sub>18</sub>からも、順番に、メッセージD316<sub>2</sub>乃至D316<sub>18</sub>を読み出して、セレクタ317に供給する。

セレクタ317には、制御部321からFIFO316<sub>1</sub>乃至316<sub>18</sub>のうち、メッセージ(データ)を読み出すFIFO(現在データが読み出されているFIFO)の選択を表す選択信号D307が供給されるとともに、枝データ格納メモリ316からメッセージデータD316<sub>1</sub>乃至D316<sub>18</sub>が供給される。セレクタ317は、選択信号D307にしたがって、FIFO316<sub>1</sub>乃至316<sub>18</sub>のうちの、現在データが読み出されているFIFOを選択し、その選択したFIFOから供給される5つのメッセージデータを、メッセージD316として、バリアブルノード計算部319に供給する。

なお、受信データ用メモリ306から供給された受信データD309に対して、



まだチェックノード演算が行われておらず、枝データ格納メモリ 316 にメッセージ D304 が格納されていない場合、バリエブルノード計算部 319 は、バリエブルノード演算に用いるメッセージ  $u_j$  を初期値に設定する。

また、バリエブルノード計算部 319 には、受信データ用メモリ 318 から 5 つの受信データ D309 (受信値  $u_{0i}$ ) が供給され、その受信データ D309 が、バリエブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> のそれぞれに 1 つずつ供給される。さらに、バリエブルノード計算部 319 には、制御部 321 から制御信号 D315 が供給され、その制御信号 D315 がバリエブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> に供給される。

- 10     バリエブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> は、メッセージ D316 と、受信データ D309 を用いて、制御信号 D315 に基づいて、式 (1) にしたがって同時に演算を行い、その演算の結果 5 つのメッセージ D319 を求める。

- 即ち、制御部 321 がバリエブルノード計算部 319 に供給する制御信号 D315 は、前述の図 11 で説明した制御信号 D107 に対応するものであり、バリエブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> は、制御信号 D309 にしたがって、セクタ 317 を介して、枝データ格納メモリ 316 から必要なメッセージ D314 (D316) を、それぞれ 1 つずつ読み出すとともに、受信データ用メモリ 318 から供給された 5 つの受信データ D309 を、それぞれ 1 つずつ読み出して、バリエブルノード演算を行い、その演算の結果 5 つのメッセージ D319 を同時に求める。
- 20

ステップ S 31 の処理後は、ステップ S 32 に進み、バリエブルノード計算部 319 は、バリエブルノード計算器 319<sub>1</sub> 乃至 319<sub>5</sub> のバリエブルノード演算の結果得られる 5 つのメッセージ D319 (メッセージ  $v_i$ ) をサイクリックシフト回路 320 に供給し、ステップ S 33 に進む。

- 25     ステップ S 33 において、サイクリックシフト回路 320 は、バリエブルノード計算部 319 から供給された 5 つのメッセージ D318 を、サイクリックシフトする (並べ替える)。

具体的には、サイクリックシフト回路 320 には、バリエブルノード計算部 319 からメッセージ D318 が供給されるとともに、制御部 321 から、そのメッセージ D318 に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D327 が供給される。サイクリックシフト回路 320 は、制御信号 D327 を元に、5つのメッセージ D327 をサイクリックシフトし、その結果をメッセージ D319 として、スイッチ 310 に供給する。

ステップ S33 の処理後は、ステップ S34 に進み、スイッチ 310 は、サイクリックシフト回路 320 から供給される 5つのメッセージ D319 を枝データ格納メモリ 311 に供給する。

具体的には、スイッチ 310 には、サイクリックシフト回路 320 からメッセージ (データ) D304 が供給されるとともに、そのメッセージ D304 が検査行列のどの行に属するかの情報を表す制御信号 D312 が供給される。スイッチ 310 は、制御信号 D312 にしたがって、メッセージ D304 を格納する FIFO を、枝データ格納メモリ 311 の FIFO 300<sub>1</sub> 乃至 300<sub>6</sub> の中から選択し、選択した FIFO に 5つのメッセージデータ D304 をまとめて順番に供給していく。

そして、枝データ格納メモリ 311 の FIFO 300<sub>1</sub> 乃至 300<sub>18</sub> は、スイッチ 310 から供給された 5つのメッセージデータ D304 をまとめて順番に格納していく。

ステップ S34 の処理後は、ステップ S35 に進み、制御部 321 は、バリエブルノード計算部 319 により、全枝数のメッセージが演算されたかどうかを判定し、全枝数のメッセージが演算されていないと判定した場合、ステップ S31 に戻り、上述した処理を繰り返す。

一方、ステップ S35 において、バリエブルノード計算部 319 は、全枝数のメッセージが演算されたと判定した場合、ステップ S36 に進み、チェックノード計算部 313 は、チェックノード演算を行う。

具体的には、チェックノード計算部 313 には、セクタ 312 を介して、5

つのメッセージ D302 が供給される。即ち、枝データ格納メモリ 3 1 1 は、ステップ S 3 4 で格納された FIFO 3 1 1<sub>1</sub> から 5 つのメッセージ D311<sub>1</sub> (メッセージ v<sub>i</sub>) を順番に読み出し、その後、FIFO 3 1 1<sub>2</sub> 乃至 3 1 1<sub>5</sub> から、順番に、メッセージデータ D311<sub>2</sub> 乃至 D311<sub>5</sub> を読み出し、セクタ 3 1 2 に供給する。

- 5     セクタ 3 1 2 には、制御部 3 2 1 から FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>5</sub> のうち、メッセージデータを読み出す FIFO (現在データが読み出されている FIFO) の選択を表す選択信号 D321 が供給されるとともに、枝データ格納メモリ 3 1 1 からメッセージデータ D311<sub>1</sub> 乃至 D311<sub>5</sub> が供給される。セクタ 3 0 1 は、選択信号 D321 にしたがって、FIFO 3 1 1<sub>1</sub> 乃至 3 1 1<sub>5</sub> のうちの、現在データが読み出されている FIFO を選択し、その選択した FIFO から供給される 5 つのメッセージデータを、メッセージ D311 として、チェックノード計算部 3 1 3 に供給する。

- 10     また、チェックノード計算部 3 1 3 には、制御部 3 2 1 から制御信号 D322 が供給される。チェックノード計算部 3 1 3 のチェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> は、制御信号 D322 に基づいて、メッセージ D302 を用いて、上述した式  
15     (7) にしたがって同時にチェックノード演算を行い、その演算結果である 5 つのメッセージ D303 (メッセージ u<sub>j</sub>) を求める。

- 即ち、制御部 3 2 1 がチェックノード計算部 3 1 3 に供給する制御信号 D322 は、前述の図 1 0 で説明した制御信号 D106 に対応するものであり、チェックノード計算器 3 1 3<sub>1</sub> 乃至 3 1 3<sub>5</sub> は、制御信号 D322 にしたがって、セクタ 3 1 2  
20     を介して、枝データ格納メモリ 3 1 1 から必要なメッセージ D311 (D312) を、それぞれ 1 つずつ読み出しながら、チェックノード演算を行い、その演算の結果 5 つのメッセージ D313 を同時に求める。

- ステップ S 3 7 の処理後は、ステップ S 3 8 に進み、チェックノード計算部 3 1 3 は、チェックノードの演算の結果得られる 5 つのメッセージ D313 をサイクリックシフト回路 3 1 4 に出力して、ステップ S 3 8 に進む。  
25

ステップ S 3 8 において、サイクリックシフト回路 3 1 4 は、チェックノード計算部 3 1 3 から供給された 5 つのメッセージ D313 を、サイクリックシフトす

る。

具体的には、サイクリックシフト回路 3 1 4 には、チェックノード計算部 3 1 3 からメッセージ D313 が供給されるとともに、制御部 3 2 1 から、そのメッセージ D313 に対応する枝が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D314 が供給される。サイクリックシフト回路 3 1 4 は、制御信号 D314 を元に、5 つのメッセージ D313 をサイクリックシフトし、その結果をメッセージ D304 として、スイッチ 3 1 5 に供給する。

ステップ S 3 8 の処理後は、ステップ S 3 9 に進み、スイッチ 3 1 5 は、サイクリックシフト回路 3 1 4 から供給される 5 つのメッセージ D304 を枝データ格納メモリ 3 1 6 に格納する。

具体的には、スイッチ 3 1 6 には、サイクリックシフト回路 3 1 4 から、5 つのメッセージ (データ) D304 が供給されるとともに、そのメッセージ (データ) D304 が検査行列のどの列に属するかの情報を表す制御信号 D324 が供給される。スイッチ 3 1 6 は、制御信号 D324 にしたがって、メッセージ D304 を格納する FIFO を、枝データ格納メモリ 3 1 6 の FIFO 3 1 6<sub>1</sub> 乃至 3 1 6<sub>18</sub> の中から選び、選んだ FIFO に 5 つのメッセージデータ D304 をまとめて順番に供給していく。

そして、枝データ格納メモリ 3 1 6 の FIFO 3 1 6<sub>1</sub> 乃至 3 1 6<sub>18</sub> は、スイッチ 3 1 6 から供給された 5 つのメッセージデータ D304 をまとめて順番に格納していく。

ステップ S 3 9 の処理後は、ステップ S 4 0 に進み、制御部 3 2 1 は、チェックノード計算部 3 1 3 により、全枝数のメッセージが演算されたかどうかを判定し、全枝数のメッセージが演算されていないと判定した場合、ステップ S 3 6 に戻り、上述した処理を繰り返す。

一方、ステップ S 4 0 において、制御部 3 2 1 は、チェックノード計算部 3 1 3 により、全枝数のメッセージが演算されたと判定した場合、処理を終了する。

なお、復号装置 300 は、復号回数だけ図 17 の復号処理を繰り返し行ない、チェックノード計算部 313 が、最後のチェックノード演算を行った場合、チェックノード演算の結果得られるメッセージ D304 が、枝データ格納メモリ 316 から、セクタ 317 を介して、上述した式 (5) の演算を行う不図示のブロックに供給される。不図示のブロックには、さらに受信データ用メモリ 306 から受信データ D309 が供給され、不図示のブロックは、メッセージ D304 と受信データ D309 を用いて、式 (5) の演算を行い、その演算結果を最終的な復号結果として出力する。

上記説明には、枝データ格納に FIFO を用いたが（枝データ格納メモリ 311 と 316 を FIFO で構成するようにしたが）、FIFO の代わりに RAM を用いても構わない。その場合、RAM には、P 個の枝情報（枝に対応するメッセージ）を同時に読み出すことの出来るビット幅と、枝総数/P のワード(word)数が必要となる。さらに、RAM への書き込みは、検査行列の情報から、書き込もうとしているデータが次に読み出される際に何番目に読み出されるかを求め、その位置に書き込む。また、RAM からの読み出しの際には、アドレスの先頭から順次データを読み出す。即ち、RAM は、メッセージデータを読み出される順番に詰めて格納し、格納位置順に読み出す。FIFO の代わりに RAM を用いると、セクタ 312 および 317 は不要になる。

なお、FIFO や RAM の物理的なビット幅が足りない場合には、複数の RAM を用いて同じ制御信号を与えることで、論理的に 1 つの RAM とみなすことができる。

ところで、図 16 A 乃至図 16 C の復号装置 300 では、チェックノード演算の結果得られるメッセージ  $u_j$  を用いて、バリアブルノード演算が行われ、その演算の結果得られるメッセージ  $v_i$  を用いて、チェックノード演算が行われるため、チェックノード演算の結果得られる枝に対応するメッセージ  $u_j$  とバリアブルノード演算の結果得られる枝に対応するメッセージ  $v_i$  をすべて格納する枝データ格納メモリ 311 と枝データ格納メモリ 316 が必要である。即ち、復号装置では、検査行列 H の “1” の数の 2 倍のメッセージを格納するために必要な

容量のメモリが必要である。

そこで、復号装置の回路規模さらに小さくするため、図16A乃至図16Cの復号装置300に比べて、さらにメモリの容量を減らした復号装置を以下に示す。

図18は、本発明を適用した図15の検査行列で表現されるLDPC符号を復号する復号装置の他の一実施の形態の構成例を示すブロック図である。

図18の復号装置400では、図16Aと図16Bの枝データ格納メモリ311が、枝データ格納メモリ311に比べて容量の小さい復号途中結果格納用メモリ410になっている。

復号装置400は、復号途中結果格納用メモリ410、サイクリックシフト回路411、5つの計算器412<sub>1</sub>乃至計算器412<sub>5</sub>からなる計算部412、復号途中結果格納用メモリ413、サイクリックシフト回路414、5つの計算器415<sub>1</sub>乃至計算器415<sub>5</sub>からなる計算部415、受信用メモリ416、および制御部417から構成される。

ここで、図19乃至図22を用いて、図18の計算部412の計算器412<sub>1</sub>乃至計算器412<sub>5</sub>、および計算部415の計算器415<sub>1</sub>乃至計算器415<sub>5</sub>と図10のチェックノード計算器101と図11のバリエブルノード計算器103との関係について説明する。

図19と図20は、前述の図10のチェックノード計算器101と図11のバリエブルノード計算器103とそれぞれ同一の図である。また、図21は、計算部412<sub>k</sub> (k=1, 2, ..., 5) の構成例を示しており、図22は、計算部415<sub>k</sub> (k=1, 2, ..., 5) の構成例を示している。

図18の復号装置400では、計算器412<sub>k</sub>がチェックノード演算を行い、計算部415<sub>k</sub>が、バリエブルノード演算をおこなうのではなく、計算器412<sub>k</sub>がチェックノード演算とバリエブルノード演算の一部を、計算器415<sub>k</sub>がバリエブルノード演算の他の一部を行う。

即ち、図21の計算器412<sub>k</sub>は、ブロックA'とブロックB'から構成されている。ブロックA'は、図19のチェックノード計算器101のチェックノード

演算を行うブロック A と同様に構成されている。また、ブロック B' は、図 20 のバリアブルノード計算器 103 の一部である、検査行列の各列の全ての枝に対応するメッセージ  $u_j$  の積算値から、求めたい枝に対応するメッセージ  $u_j$  を減算するブロック B と同様に構成されている。一方、図 22 の計算器 415<sub>k</sub> は、ブロック C' から構成されている。ブロック C' は、図 20 のバリアブルノード計算器 103 の他の一部である、検査行列の各列の枝に対応するメッセージ  $u_j$  を積算し、その積算値に受信値  $u_{oi}$  を加算するブロック C と同様に構成されている。

そして、図 21 の計算器 412<sub>k</sub> は、ブロック A とブロック B による演算の結果、即ち、チェックノード演算とバリアブルノード演算の一部を行った復号途中結果  $u_j$  を復号途中結果格納用メモリ 413 に供給し、図 22 の計算器 415<sub>k</sub> は、バリアブルノード演算の他の一部を行った復号途中結果  $v$  を復号途中結果格納用メモリ 410 に供給する。

従って、図 18 の復号装置 400 は、計算器 412<sub>k</sub> の演算と計算器 415<sub>k</sub> の演算とを交互に行うことにより、チェックノード演算とバリアブルノード演算を行い、復号を行うことができる。

なお、図 22 の計算器 412<sub>k</sub> では、復号途中結果格納用メモリ 413 に格納されている求めたい枝に対応する復号途中結果  $u_j$  を用いて、ブロック B で、計算器 415<sub>k</sub> の演算の結果得られる復号途中結果  $v$  から、求めたい枝に対応する復号途中結果  $u_j$  を減算するので、図 20 の FIFO メモリ 155 が必要ない。

次に、計算器 412<sub>k</sub> で行われる演算と、計算器 415<sub>k</sub> で行われる演算について、式を用いて説明する。

具体的には、計算部 412 は、上述した式 (7) と、以下に表す式 (8) にしたがう第 1 の演算を行い、その第 1 の演算の結果である復号途中結果  $u_j$  を復号途中結果格納用メモリ 410 に供給して格納させる。計算部 415 は、上述した式 (5) にしたがう第 2 の演算を行い、その第 2 の演算の結果である復号途中結果  $v$  を復号途中結果格納用メモリ 410 に供給して格納させる。

$$v_i = v - u_{dv} \quad \dots (8)$$

なお、式(8)の $u_{dv}$ は、検査行列Hのi列のメッセージを求めようとする枝からのチェックノード演算の途中結果(ここでは、チェックノード演算結果そのもの)を表している。即ち、 $u_{dv}$ は、求めたい枝に対応する復号途中結果である。

- 即ち、上述した式(5)にしたがう第2の演算の結果得られる復号途中結果v
- 5 は、受信値 $u_{0i}$ と検査行列Hのi列の各行の1に対応するすべての枝からのチェックノード演算の復号途中結果 $u_j$ とを加算したものであるので、上述した式(7)に用いられる値 $v_i$ は、式(5)にしたがう第2の演算の結果得られる復号途中結果vから、検査行列Hのi列の、各行の1に対応する枝からのチェックノード演算の復号途中結果 $u_j$ のうち、メッセージを求めようとする枝からの
- 10 チェックノード演算の復号途中結果 $u_{dv}$ を引いた値となる。つまり、式(7)の演算に用いられる値 $v_i$ を求める式(1)の演算は、上述した式(5)と式(8)を組み合わせた演算である。

- 従って、復号装置400では、計算部412による式(7)および式(8)にしたがう第1の演算と、計算部415による式(5)にしたがう第2の演算とが
- 15 交互に行われ、計算部415が、最後の第2の演算の結果を復号結果として出力することにより、LDPC符号の繰り返し復号を行うことができる。

なお、ここでは、式(7)と式(8)にしたがう第1の演算結果を、復号途中結果 $u_j$ を復号途中結果 $u_j$ と記載するが、この復号途中結果 $u_j$ は、式(7)のチェックノード演算結果 $u_j$ に等しい。

- 20 また、第2の演算により求められる式(5)のvは、式(1)のバリエーションノード演算結果 $v_i$ に対して、メッセージを求めようとする枝からのチェックノード演算結果 $u_j$ を加算したものであるから、検査行列Hの1列(1つのバリエーションノード)に対して、1つだけ求められる。

- 復号装置400では、計算部412が、計算部415による第2の演算の結果
- 25 である検査行列Hの列に対応する復号途中結果v(第2の復号途中結果)を用いて、第1の演算を行い、その演算の結果得られる検査行列Hのi列の、各行の1に対応する枝のメッセージ(各チェックノードが各枝に出力するメッセージ)



の枝からのチェックノード演算の復号途中結果  $u_j$  (第1の復号途中結果) を復号途中結果格納用メモリ 413 に格納する。従って、復号途中結果格納用メモリ 413 の容量は、チェックノード演算の結果を格納する図16AとCの枝データ格納メモリ 316 と同様に、検査行列の1の数(全枝数)とメッセージの量子化  
5 ビット数とを乗算した値となる。一方、計算部 415 は、計算部 412 による第1の演算の結果である検査行列 H の  $i$  列の、各行の“1”に対応する復号途中結果  $u_j$  と受信値  $u_{0i}$  を用いて、第2の演算を行い、その演算の結果得られる  $i$  列に対応する復号途中結果  $v$  を復号途中結果格納用メモリ 410 に格納する。従って、復号途中結果格納用メモリ 410 に必要な容量は、検査行列の“1”の数  
10 より少ない検査行列の列数、即ち、LDPC 符号の符号長と復号途中結果  $v$  の量子化ビット数とを乗算した値となる。

従って、検査行列 H における1が疎らな LDPC 符号を復号する復号装置 400 では、図16AとBの枝データ格納メモリ 311 に比べて、復号途中結果格納用メモリ 410 のメモリの容量を削減することができ、これにより、復号装置 40  
15 0 の回路規模を小さくすることができる。

さらに、復号装置 400 では、計算部 415 が、式(5)にしたがう第2の演算を行うので、復号装置 400 は、図16A乃至図16Cの復号装置 300 において最終的な復号結果を演算する式(5)の演算を行う不図示のブロックを有する必要がなく、図16A乃至図16Cの復号装置 300 に比べて、図18の復号  
20 装置の回路規模を小さくすることができる。

以下、図18の復号装置 400 の各部の動作について詳細に説明する。

復号途中結果格納用メモリ 410 には、計算部 415 から、計算部 415 による第2の演算の結果である検査行列の5つの列に対応する5つの復号途中結果 D415 が供給され、復号途中結果格納用メモリ 410 は、計算部 415 から供給  
25 された5つの復号途中結果 D415 を、第1アドレスから順に格納(記憶)する。

即ち、復号途中結果格納用メモリ 410 の第1アドレスには、検査行列の列に対応する復号途中結果のうち、第1列目から第5列目の復号途中結果  $v$  が格納

される。そして、同様に、第2アドレスには、第6列目から第10列目の復号途中結果  $v$  が格納され、第3アドレスには、第11列目から第15列目の復号途中結果が格納される。以後、同様に、第16列目から第90列目までの復号途中結果  $v$  が、5個ずつ、第4アドレスから第18アドレスまで格納され、計90

- 5 個の復号途中結果  $v$  が復号途中結果格納用メモリ410に格納される。従って、復号途中結果格納用メモリ410のワード (word) 数は、図15の検査行列  $H$  の列数 (LDPC 符号の符号長) である90を、同時に読み書きする復号途中結果の数である5で割り算した18となる。

- また、復号途中結果格納用メモリ410は、既に格納してある復号途中結果  
10 D415 から、後段の計算部412が求めようとする復号途中結果  $u_j$  の対応する検査行列  $H$  の行において“1”になっている復号途中結果  $v$  を5つ同時に読み出し、復号途中結果 D410 として、サイクリックシフト回路411に供給する。

- なお、復号途中結果格納用メモリ410は、例えば、5つの復号途中結果を同時に読み書き可能なシングルポート RAM で構成される。また、復号途中結果格  
15 納用メモリ410には、計算部415の第2の演算により演算された列に対応する復号途中結果  $v$  が格納されるので、復号途中結果格納用メモリ410に格納されるデータ量、即ち、復号途中結果格納用メモリ410に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列  $H$  の列数 (LDPC 符号の符号長) との乗算値である。

- 20 サイクリックシフト回路411には、復号途中結果格納用メモリ410から5つの復号途中結果 D410 が供給されるとともに、制御部417から、その復号途中結果 D410 に対応する検査行列の1が、検査行列において元となる単位行列などを幾つサイクリックシフトであるかの情報 (Matrix データ) を表す制御信号 D619 が供給される。サイクリックシフト回路611は、制御信号 D619 を元に、  
25 5つの復号結果 D410 を並べ替えるサイクリックシフトを行い、その結果を復号途中結果 D411 として、計算部412に供給する。

計算部412は、5つの計算器412<sub>1</sub>乃至412<sub>5</sub>からなる。計算部412に

は、サイクリックシフト回路 4 1 1 から、計算部 4 1 5 による第 2 の演算の結果得られた 5 つの復号途中結果 D411 (第 2 の復号途中結果)  $v$  が供給されるとともに、復号途中結果格納用メモリ 4 1 3 から、前回、計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> による第 1 の演算の結果得られた 5 つの復号途中結果 D413 (第 1 の復号途中結果)  $u_j$  が供給され、その 5 つの復号途中結果 D411 と 5 つの復号途中結果 D413 が、計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> にそれぞれ供給される。また、計算部 4 1 2 には、制御部 4 1 7 から制御信号 D419 が供給され、その制御信号 D419 が、計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> に供給される。なお、制御信号 D419 は、5 つの計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> に共通の信号である。

10 計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> は、それぞれ復号途中結果 D411 と復号途中結果 D413 を用いて、式 (7) と式 (8) にしたがって第 1 の演算を行い、復号途中結果 D412 ( $v_i$ ) を求める。計算部 4 1 2 は、計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> による演算の結果得られる検査行列の 5 つの 1 に対応する 5 つの復号途中結果 D412 を復号途中結果格納用メモリ 4 1 3 に供給する。

15 復号途中結果格納用メモリ 4 1 3 は、例えば、5 つの復号途中結果を同時に読み書き可能な、2 つのシングルポート RAM から構成される。復号途中結果格納用メモリ 4 1 3 には、計算部 4 1 2 から 5 つの復号途中結果 D412 が供給されるとともに、制御部 4 1 7 から復号途中結果 4 1 3 の読み書きを制御する制御信号 D420 が供給される。

20 復号途中結果格納用メモリ 4 1 3 は、制御信号 D420 に基づいて、計算部 4 1 2 から供給される 5 つの復号途中結果 D412 をまとめて格納すると同時に、既に格納してある 5 つの復号途中結果 D412 を読み出し、復号途中結果 D413 として、計算部 4 1 2 とサイクリックシフト回路 4 1 4 に供給する。即ち、復号途中結果格納用メモリ 4 1 3 は、計算部 4 1 2 とサイクリックシフト回路 4 1 4 に供給する復号途中結果 D413 の読み出しと、計算部 4 1 2 から供給される復号途中結果 D412 の書き込みとを、同時に行う。

25 なお、復号途中結果格納用メモリ 4 1 3 には、計算部 4 1 2 の第 1 の演算によ

り演算された検査行列  $H$  の  $i$  列の、各行の 1 に対応する枝からのチェックノード演算の復号途中結果  $u_j$  が格納されるので、復号途中結果格納用メモリ 4 1 3 に格納されるデータ量、即ち、復号途中結果格納用メモリ 4 1 3 に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列の 1 の数との乗算値となる。

サイクリックシフト回路 4 1 4 には、復号途中結果格納用メモリ 4 1 3 から 5 つの復号途中結果 D413 (復号途中結果  $u_j$ ) が供給されるとともに、制御部 4 1 7 から、その復号途中結果 D413 に対応する検査行列の 1 が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報

10 (Matrix データ) を表す制御信号 D421 が供給される。サイクリックシフト回路 4 1 4 は、制御信号 D421 を元に、5 つの復号途中結果 D413 を並べ替えるサイクリックシフトを行い、その結果を復号途中結果 D414 として、計算部 4 1 5 に供給する。

計算部 4 1 5 は、5 つの計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> からなる。計算部 4 1 5 に  
15 は、サイクリックシフト回路 4 1 4 から 5 つの復号途中結果 D414 が供給され、その復号途中結果 D414 が、計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> のそれぞれに供給される。また、計算部 4 1 5 には、受信用メモリ 4 1 7 から 5 つの受信データ D417 (LDPC 符号) が供給され、その受信データ D417 が、計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> のそれぞれに供給される。さらに、計算部 4 1 7 には、制御部 4 1 7 から制御信号 D422 が供給され、その制御信号 D422 が計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> に供給さ  
20 される。なお、制御信号 D422 は、5 つの計算器 4 1 7<sub>1</sub> 乃至 4 1 7<sub>5</sub> に共通の信号である。

計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> は、それぞれ復号途中結果 D414 と受信データ D417 とを用いて、式 (5) にしたがって、それぞれ第 2 の演算を行い、復号途中結果  
25 D415 を求める。計算部 4 1 5 は、計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> の第 2 の演算の結果得られる 5 つの復号途中結果 D415(v) を、復号途中結果格納用メモリ 4 1 0 に供給する。また、計算部 4 1 5 は、いま行う演算が最後の第 2 の演算である場合、

その演算の結果得られる 5 つの復号途中結果 D415 を、最終的な復号結果として出力する。

受信用メモリ 416 は、通信路を通して受信した受信値（符号ビット）D416 から計算した符号ビットの 0 らしさの値である受信 LLR（対数尤度比）を、受信  
5 データ D417 として格納する。

即ち、受信用メモリ 416 の第 1 のアドレスには、検査行列の列に対応する受信データ D417 のうち、検査行列の第 1 列目から第 5 列目までに対応する受信データ D417 が格納される。そして、第 2 のアドレスには、検査行列の第 6 列目から第 10 列目までに対応する受信データ D417 が格納され、第 3 アドレスには、  
10 検査行列の第 11 列目から第 16 列目までに対応する受信データ D417 が格納される。以後、同様に、第 4 アドレスから第 18 アドレスまでに、検査行列の第 17 列目から第 90 列目までに対応する受信データ D417 が、5 つずつ格納される。

そして、受信用メモリ 616 は、既に格納している受信データ D417 を、バリアブルノード演算に必要となる順番に 5 つずつ読み出し、計算部 415 に供給す  
15 る。

なお、受信用メモリ 416 は、例えば、5 つの受信データを同時に読み書き可能なシングルポート RAM から構成される。また、受信用メモリ 416 に格納されるデータ量、即ち、受信用メモリ 315 に必要とされる記憶容量は、LDPC 符号の符号長と、受信データの量子化ビット数との乗算値である。さらに、受信用  
20 メモリ 416 のワード（word）数は、LDPC 符号の符号長、即ち、検査行列の列数である 90 を、同時に読み出す受信データ D417 の数である 5 で割り算した値の 18 である。

制御部 417 は、制御信号 D418 をサイクリックシフト回路 411 に、制御信号 D419 を計算部 412 に供給することにより、それぞれを制御する。また、制御部 417 は、制御信号 D420 を復号途中結果格納用メモリ 413 に、制御信号  
25 D421 をサイクリックシフト回路 414 に、制御信号 D421 を計算部 415 にそれぞれ供給することにより、それぞれを制御する。

復号途中結果格納用メモリ 410、サイクリックシフト回路 411、計算部 412、復号途中結果格納用メモリ 413、サイクリックシフト回路 414、計算部 415 の順で、データが一巡することで、復号装置 400 は、1 回の復号を行うことができる。復号装置 400 では、所定の回数だけ繰り返して復号が行われた後、計算部 415 による第 2 の演算の結果である復号途中結果 D415 が、最終的な復号結果として出力される。

図 21 は、図 18 の計算部 412 の計算器 412<sub>1</sub> の構成例を示すブロック図である。

なお、図 21 では、計算器 412<sub>1</sub> について説明するが、計算器 412<sub>2</sub> 乃至計算器 412<sub>5</sub> も同様に構成される。

また、図 21 では、前回の計算部 412 による第 1 の演算の結果得られる各復号途中結果 ( $u_{dv}$ ) が符号ビットを合わせて合計 6 ビット (bit) に量子化され、計算器 415 による第 2 の演算の結果得られる各復号途中結果 ( $v$ ) が 9 ビットに量子化されているものとして、計算器 412<sub>1</sub> を表している。さらに、図 21 の計算器 412<sub>1</sub> には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

図 21 の計算器 412<sub>1</sub> は、制御部 417 から供給される制御信号 D419 に基づいて、復号途中結果格納用メモリ 413 から 1 つずつ読み込まれる、前回の計算部 412 による第 1 の演算の結果得られた復号途中結果 D413 ( $u_{dv}$ ) と、サイクリックシフト回路 411 から 1 つずつ読み込まれる復号途中結果 D411 ( $v$ ) とを用いて、式 (7) と式 (8) にしたがう第 1 の演算を行う。

即ち、計算器 412<sub>1</sub> には、サイクリックシフト回路 411 から供給される 5 つの 9 ビットの復号途中結果 D411 ( $v$ ) のうちの、1 つの復号途中結果 D411 が供給されるとともに、復号途中結果格納用メモリ 413 から供給される、前回の計算部 412 による演算の結果である 5 つの 6 ビットの復号途中結果 D413 ( $u_j$ ) のうちの、前回の計算部 412 による演算の結果である 1 つの復号途中結果

D413 が供給され、その 9 ビットの復号途中結果 D411 ( $v$ ) と 6 ビットの復号途中結果 D413 ( $u_{dv}$ ) が、減算器 4 3 1 に供給される。また、計算器 4 1 2<sub>1</sub>には、制御部 4 1 7 から制御信号 D419 が供給され、その制御信号 D419 がセクタ 4 3 5 とセクタ 4 4 2 に供給される。

- 5 減算器 4 3 1 は、9 ビットの復号途中結果 D411 ( $v$ ) から 6 ビットの復号途中結果 D413 ( $u_j$ ) を減算し、その 6 ビットの減算値 D431 を出力する。即ち、減算器 4 3 1 は、式 (8) にしたがって演算を行い、その演算の結果である減算値 D431 ( $v_i$ ) を出力する。

- 10 減算器 4 3 1 により出力された 6 ビットの減算値 D431 のうち、最上位ビットの正負を示す符号ビット D432 ( $\text{sign}(v_i)$ ) が EXOR 回路 4 4 0 に供給され、下位 5 ビットの絶対値 D433 ( $|v_i|$ ) が LUT 4 3 2 に供給される。

LUT 4 3 2 は、絶対値 D433 ( $|v_i|$ ) に対して、式 (7) における  $\phi(|v_i|)$  の演算を行った 5 ビットの演算結果 D434 ( $\phi(|v_i|)$ ) を読み出し、加算器 4 3 3 と FIFO メモリ 4 3 8 に供給する。

- 15 加算器 4 3 3 は、演算結果 D434 ( $\phi(|v_i|)$ ) とレジスタ 4 3 4 に格納されている 9 ビットの値 D435 とを加算することにより、演算結果 D434 を積算し、その結果得られる 9 ビットの積算値をレジスタ 4 3 4 に再格納する。なお、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 から求められた絶対値 D433 ( $|v_i|$ ) に対する演算結果が積算された場合、レジスタ 4 3 4 はリセットされる。
- 20

- 検査行列の 1 行に亘る復号途中結果 D411 が 1 つずつ読み込まれ、レジスタ 4 3 4 に 1 行分の演算結果 D434 が積算された積算値が格納された場合、制御部 4 1 7 から供給される制御信号 D419 は、0 から 1 に変化する。例えば、行の重み (row weight) が「9」である場合、制御信号 D419 は、1 から 8 クロック目までは、「0」となり、9 クロック目では「1」となる。
- 25

制御信号 D419 が「1」の場合、セクタ 4 3 5 は、レジスタ 4 3 4 に格納されている値、即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果

D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  が積算された 9 ビットの値 D435 ( $i = 1$  から  $i = d_0$  までの  $\sum \phi(|v_i|)$ ) を選択し、値 D436 として、レジスタ 436 に出力して格納させる。レジスタ 436 は、格納している値 D436 を、9 ビットの値 D437 として、セクタ 435 と加算器 437 に供給する。制御信号 D419 が「0」の場合、セクタ 435 は、レジスタ 436 から供給された値 D437 を選択し、レジスタ 436 に出力して再格納させる。即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  が積算されるまで、レジスタ 436 は、前回積算された  $\phi(|v_i|)$  を、セクタ 435 と加算器 437 に供給する。

- 10 一方、FIFO メモリ 438 は、レジスタ 436 から新たな値 D437 ( $i = 1$  から  $i = d_0$  までの  $\sum \phi(|v_i|)$ ) が出力されるまでの間、LUT 432 が出力した演算結果 D434 ( $\phi(|v_i|)$ ) を遅延し、5 ビットの値 D438 として減算器 437 に供給する。減算器 437 は、レジスタ 436 から供給された値 D437 から、FIFO メモリ 438 から供給された値 D438 を減算し、その減算結果を、5 ビットの減算値 D439 として LUT 439 に供給する。即ち、減算器 437 は、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  の積算値から、求めたい枝に対応する復号途中結果、即ち、検査行列の所定の 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた  $\phi(|v_i|)$  を減算して、その減算値 ( $i = 1$  から  $i = d_0 - 1$  までの  $\sum \phi(|v_i|)$ ) を減算値 D439 として LUT 439 に供給する。

LUT 439 は、減算値 D439 ( $i = 1$  から  $i = d_0 - 1$  までの  $\sum \phi(|v_i|)$ ) に対して、式 (7) における  $\phi^{-1}(\sum \phi(|v_i|))$  の演算を行った 5 ビットの演算結果 D440 ( $\phi^{-1}(\sum \phi(|v_i|))$ ) を出力する。

- 25 以上の処理と並行して、EXOR 回路 440 は、レジスタ 441 に格納されている 1 ビットの値 D442 と符号ビット D432 との排他的論理和を演算することにより、符号ビットどうしの乗算を行い、1 ビットの乗算結果 D441 をレジスタ 441 に再格納する。なお、検査行列の 1 行に亘る全ての 1 に対応する復号途中結



果 D411 から求められた符号ビット D432 が乗算された場合、レジスタ 4 4 1 はリセットされる。

検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D411 から求められた符号ビット D432 が乗算された乗算結果 D441 ( $i = 1$  から  $d_0$  までの

- 5  $\prod \text{sign}(v_i)$ ) がレジスタ 4 4 1 に格納された場合、制御部 4 1 7 から供給される制御信号 D419 は、「0」から「1」に変化する。

制御信号 D419 が「1」の場合、セクタ 4 4 2 は、レジスタ 4 4 1 に格納されている値、即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果

D411 から求められた符号ビット D432 が乗算された値 D442 ( $i = 1$  から  $i = d_0$

- 10 までの  $\prod \text{sign}(v_i)$ ) を選択し、1 ビットの値 D443 としてレジスタ 4 4 3 に出力して格納させる。レジスタ 4 4 3 は、格納している値 D443 を、1 ビットの値 D444 としてセクタ 4 4 2 と EXOR 回路 4 4 5 に供給する。制御信号 D419 が「0」の場合、セクタ 4 4 2 は、レジスタ 4 4 3 から供給された値 D444 を選択し、レジスタ 4 4 3 に出力して再格納させる。即ち、検査行列の 1 行に亘る  
15 全ての 1 に対応する復号途中結果 D411 (復号途中結果  $v$ ) から求められた符号ビット D432 が乗算されるまで、レジスタ 4 4 3 は、前回格納した値を、セクタ 4 4 2 と EXOR 回路 4 4 5 に供給する。

一方、FIFO メモリ 4 4 4 は、レジスタ 4 4 3 から新たな値 D444 ( $i = 1$  から  $i = d_0$  までの  $\prod \text{sign}(v_i)$ ) が EXOR 回路 4 4 5 に供給されるまでの間、符号ビ

- 20 ット D432 を遅延し、1 ビットの値 D445 として EXOR 回路 4 4 5 に供給する。

EXOR 回路 4 4 5 は、レジスタ 4 4 3 から供給された値 D444 と、FIFO メモリ 4 4 4 から供給された値 D445 との排他的論理和を演算することにより、値 D444

を、値 D445 で除算し、1 ビットの除算結果を除算値 D446 として出力する。即ち、EXOR 回路 4 4 5 は、検査行列の 1 行に亘る全ての 1 に対応する復号途中結

- 25 果 D411 から求められた符号ビット D432 ( $\text{sign}(v_i)$ ) の乗算値を、検査行列の所定の 1 に対応する復号途中結果 D411 から求められた符号ビット D432

( $\text{sign}(v_i)$ ) で除算して、その除算値 ( $i = 1$  から  $i = d_0 - 1$  までの

$\Pi \text{sign}(v_i))$  を除算値 D446 として出力する。

計算器 4 1 2<sub>1</sub> では、LUT 4 3 9 から出力された 5 ビットの演算結果 D440 を下位 5 ビットとするとともに、EXOR 回路 4 4 5 から出力された 1 ビットの除算値 D446 を最上位ビットとする合計 6 ビットが復号途中結果 D412 (復号途中結果 5  $u_j$ ) として出力される。

以上のように、計算器 4 1 2<sub>1</sub> では、式 (7) と式 (8) の演算が行われ、復号途中結果  $u_j$  が求められる。

なお、図 1 5 の検査行列の行の重みの最大は 9 であるため、即ち、計算器 4 1 2<sub>1</sub> に供給される復号途中結果 D411 ( $v$ ) と復号途中結果 D413 ( $u_{dv}$ ) の最大数は 9 であるため、計算器 4 1 2<sub>1</sub> は、9 個の復号途中結果 D411 から求められる 9 個の演算結果 D434 ( $\phi(|v_i|)$ ) を遅延させる FIFO メモリ 4 3 8 と、9 個の符号ビット D432 を遅延させる FIFO メモリ 4 4 4 を有している。行の重みが 9 未満の行のメッセージを計算するときには、FIFO メモリ 4 3 8 と FIFO メモリ 4 4 4 における遅延量が、その行の重みの値に減らされる。

15 図 2 2 は、図 1 8 の計算部 4 1 5 の計算器 4 1 5<sub>1</sub> の構成例を示すブロック図である。

なお、図 2 2 では、計算器 4 1 5<sub>1</sub> について説明するが、計算器 4 1 5<sub>2</sub> 乃至計算器 4 1 5<sub>5</sub> も同様に構成される。

また、図 2 2 では、計算器 4 1 2 による第 1 の演算の結果得られる各復号途中 20 結果 ( $u_j$ ) が符号ビットを合わせて合計 6 ビットに量子化されているものとして、計算器 4 1 5<sub>1</sub> を表している。さらに、図 2 2 の計算器 4 1 5<sub>1</sub> には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

図 2 2 の計算器 4 1 5<sub>1</sub> は、制御部 4 1 7 から供給される制御信号 D422 に基 25 づいて、受信用メモリ 4 1 6 から 1 つずつ読み込まれる受信データ D417 (受信値  $u_{oi}$ ) と、サイクリックシフト回路 4 1 4 から 1 つずつ読み込まれる復号途中結果 D414 ( $u_j$ ) とを用いて、式 (5) にしたがう第 2 の演算を行う。

即ち、計算器 4 1 5<sub>1</sub>では、サイクリックシフト回路 4 1 4 から、検査行列の各行の 1 に対応する 6 ビットの復号途中結果 D414 (復号途中結果  $u_j$ ) が 1 つずつ読み込まれ、その復号途中結果 D414 が、加算器 4 7 1 に供給される。また、計算器 4 1 5<sub>1</sub>では、受信用メモリ 4 1 6 から 6 ビットの受信データ D417 が 1 つずつ読み込まれ、加算器 4 7 5 に供給される。さらに、計算器 4 1 5<sub>1</sub>には、制御部 4 1 7 から制御信号 D422 が供給され、その制御信号 D422 は、セクタ 4 7 3 に供給される。

加算器 4 7 1 は、復号途中結果 D414 (復号途中結果  $u_j$ ) とレジスタ 4 7 2 に格納されている 9 ビットの値 D471 とを加算することにより、復号途中結果 D414 を積算し、その結果得られる 9 ビットの積算値を、レジスタ 4 7 2 に再格納する。なお、検査行列の 1 列に亘る全ての 1 に対応する復号途中結果 D414 が積算された場合、レジスタ 4 7 2 はリセットされる。

検査行列の 1 列に亘る復号途中結果 D414 が 1 つずつ読み込まれ、レジスタ 4 7 2 に 1 列分の復号途中結果 D414 が積算された値が格納された場合、制御部 4 1 7 から供給される制御信号 D422 は、「0」から「1」に変化する。例えば、列の重みが「5」である場合、制御信号 D422 は、1 から 4 クロック目までは「0」となり、5 クロック目では「1」となる。

制御信号 D422 が「1」の場合、セクタ 4 7 3 は、レジスタ 4 7 2 に格納されている値、即ち、検査行列の 1 列に亘る全ての枝からの復号途中結果 D414 (復号途中結果  $u_j$ ) が積算された 9 ビットの値 D471 ( $j = 1$  から  $d_v$  までの  $\sum u_j$ ) を選択し、レジスタ 4 7 4 に出力して格納させる。レジスタ 4 7 4 は、格納している値 D471 を、9 ビットの値 D472 として、セクタ 4 7 1 と加算器 4 7 5 に供給する。制御信号 D422 が「0」の場合、セクタ 4 7 3 は、レジスタ 4 7 4 から供給された値 D472 を選択し、レジスタ 4 7 4 に出力し再格納させる。即ち、検査行列の 1 列に亘る全ての枝からの復号途中結果 D414 (復号途中結果  $u_j$ ) が積算されるまで、レジスタ 4 7 4 は、前回積算された値を、セクタ 4 7 3 と加算器 4 7 5 に供給する。

加算器 4 7 5 は、9 ビットの値 D472 と、受信用メモリ 4 1 6 から供給された 6 ビットの受信データ D417 とを加算して、その結果得られる 6 ビットの値を復号途中結果 D415（復号途中結果  $v$ ）として出力する。

5 以上のように、計算器 4 1 5<sub>1</sub>では、式（5）の演算が行われ、復号途中結果  $v$  が求められる。

なお、図 8 の検査行列の列の重みの最大は 5 であるため、即ち、計算器 4 1 5<sub>1</sub>に供給される復号途中結果  $u_j$  の最大数は 5 であるため、計算器 4 1 5<sub>1</sub>は、6 ビットの復号途中結果  $u_j$  を最大 5 個加算する。従って、計算器 4 1 5<sub>1</sub>の出力は、9 ビットの値となっている。

10 図 2 3 は、図 1 8 の復号途中結果格納用メモリ 4 1 3 の構成例を示すブロック図である。

復号途中結果格納用メモリ 4 1 3 は、スイッチ 5 0 1 と 5 0 4、および 2 つのシングルポート RAM である復号途中結果格納用 RAM 5 0 2 と 5 0 3 から構成される。

15 この復号途中結果格納用メモリ 4 1 3 の各部について詳細に説明する前に、まず、復号途中結果格納用 RAM 5 0 2 と 5 0 3 へのデータの格納方法について説明する。

復号途中結果格納用 RAM 5 0 2 と 5 0 3 は、計算部 4 1 2 による第 1 の演算の結果得られ、スイッチ 5 0 1 を介して供給された復号途中結果 D412 を格納する。

20 具体的には、復号途中結果格納用 RAM 5 0 2 の第 1 アドレスから第 9 アドレスには、図 1 5 の検査行列  $H$  の第 1 行目から第 5 行目までの 1 に対応する復号途中結果 D412 (D501) が、各行ともに横方向（列方向）に詰めた形に（0 を無視した形で）格納される。

即ち、第  $j$  行第  $i$  列を、 $(j, i)$  と表すこととすると、復号途中結果格納用 RAM  
25 5 0 2 の第 1 アドレスには、図 1 5 の検査行列の構成行列である  $(1, 1)$  から  $(5, 5)$  の  $5 \times 5$  の単位行列の 1 に対応するデータが、第 2 アドレスには、図 1 5 の検査行列の構成行列である  $(1, 21)$  から  $(5, 25)$  のシフト行列（ $5 \times 5$  の単位行

列を右方向に3つだけサイクリックシフトしたシフト行列)の1に対応するデータが格納される。第3アドレスから第8アドレスも同様に図15の検査行列の構成行列と対応づけてデータが格納される。そして、第9アドレスには、検査行列の(1,86)から(5,90)のシフト行列(5×5の単位行列のうちの1行目の1を0に置き換えて1つだけ左にサイクリックシフトしたシフト行列)の1に対応するデータが格納される。ここで、図15の検査行列の(1,86)から(5,90)のシフト行列においては、1行目に1がないため、第9アドレスにはデータが格納されない。

復号途中格納用RAM502の第10アドレスから第18アドレスには、図15の検査行列の第11行目から第15行目までの1に対応するデータが格納される。即ち、第10アドレスには、検査行列の(11,6)から(15,10)の5×5の単位行列を右に3つだけサイクリックシフトした行列の1に対応するデータが格納され、第11アドレスには、検査行列の(11,11)から(15,15)の和行列(5×5の単位行列と、5×5の単位行列を右に3つだけサイクリックシフトしたシフト行列との和である和行列)を構成するシフト行列の1に対応するデータが格納される。また、第12アドレスには、検査行列の(11,6)から(15,10)の和行列を構成する単位行列の1に対応するデータが格納される。以下、第13アドレスから第18アドレスについても、検査行列に対応づけてデータが格納される。

即ち、重みが2以上の構成行列については、その構成行列を、重みが1であるP×Pの単位行列、そのコンポーネントである1のうち1個以上が0になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列の1の位置に対応するデータ(単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージの復号途中結果)は、同一アドレスに格納される。

同様に、復号途中格納用RAM502の第19アドレスから第27アドレスには、図15の検査行列に対応づけて、第21行目から第25行目までの1に対応する

データが格納される。即ち、復号途中結果格納用 RAM 5 0 2 のワード数は、27である。

復号途中結果格納用 RAM 5 0 3 の第1アドレスから第9アドレスには、図15の検査行列 H の第6行目から第10行目までの1に対応する復号途中結果

- 5 D412(D502)が、各行ともに横方向(列方向)に詰めた形に(0を無視した形で)格納される。

即ち、復号途中結果格納用 RAM 5 0 3 の第1アドレスには、検査行列の構成行列である(6,1)から(10,5)の和行列(5×5の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2

10 のシフト行列の和である和行列)を構成する第1のシフト行列の1に対応するデータが、第2アドレスには、検査行列の構成行列である(6,1)から(10,5)の和行列を構成する第2のシフト行列の1に対応するデータが格納される。以下、第3アドレスから第9アドレスも同様に検査行列の構成行列と対応づけてデータが格納される。

- 15 同様に、復号途中格納用 RAM 5 0 3 の第10アドレスから第18アドレスには、図15の検査行列の第16行目から第20行目までの1に対応するデータが、第19アドレスから第27アドレスには、検査行列の第26行目から第30行目までの1に対応するデータが、図15の検査行列に対応づけて格納される。即ち、復号途中結果格納用 RAM 5 0 3 のワード数は、27である。

- 20 上述したように、復号途中結果格納用 RAM 5 0 2 と 5 0 3 のワード(word)数は、27である。即ち、ワード数は、検査行列の行の重み(row weight)の9と行数の30とを乗算し、その乗算結果(検査行列の1の数)を、同時に読み出す復号途中結果 D501 の数の5で除算し、さらに、復号途中結果格納用メモリ 4 1 3 が有する復号途中結果格納用 RAM の個数の2で除算した値となる。

- 25 以下、図23の復号途中結果格納用メモリ 4 1 3 の各部の動作について詳細に説明する。

復号途中結果格納用メモリ 4 1 3 には、計算部 4 1 2 により第1の演算が行わ

れる場合、計算部 4 1 2 から第 1 の演算の結果得られる復号途中結果 D412

( $u_j$ ) が供給され、その復号途中結果 D412 が復号途中結果格納用 RAM 5 0 2 または復号途中結果格納用 RAM 5 0 3 のうちの一方の所定のアドレスに書き込まれると同時に、他方から、前回の計算部 4 1 2 による第 1 の演算の結果得られた復  
5 号途中結果 D412 ( $u_j$ ) が読み出され、計算部 4 1 2 に出力される。一方、計算部 4 1 5 により第 2 の演算が行われる場合、復号途中結果格納用メモリ 4 1 3 は、復号途中結果格納用 RAM 5 0 2 または復号途中結果格納用 RAM 5 0 3 に書き込みを行わず、どちらか一方の RAM の所定のアドレスから復号途中結果を読み出して、サイクリックシフト回路 4 1 4 に供給する。

- 10     スイッチ 5 0 1 には、計算部 4 1 2 から 5 つの復号途中結果 D412 が供給されるとともに、その復号途中結果 D412 を書き込むメモリとして、復号途中結果格納用 RAM 5 0 2 または復号途中結果格納用 RAM 5 0 3 の一方の選択を表す制御信号 D420<sub>1</sub> が制御部 4 1 7 から供給される。スイッチ 5 0 1 は、制御信号 D420<sub>1</sub> に基づいて、復号途中結果格納用 RAM 5 0 2 または復号途中結果格納用 RAM 5 0 3  
15     の一方を選択し、その選択した一方に、5 つの復号途中結果 D412 を供給する。

- 復号途中結果格納用 RAM 5 0 2 には、スイッチ 5 0 1 から 5 つの復号途中結果 D412 が、復号途中結果 D501 として供給されるとともに、制御部 4 1 7 からアドレスを表す制御信号 D420<sub>2</sub> が供給される。復号途中結果格納用 RAM 5 0 2 は、制御信号 D420<sub>2</sub> が表すアドレスに既に格納されている前回の計算部 4 1 2 による第  
20     1 の演算の結果得られた 5 つの復号途中結果 D501 を読み出し、復号途中結果 D503 としてスイッチ 5 0 4 に供給する。また、復号途中結果格納用 RAM 5 0 2 は、制御信号 D420<sub>2</sub> が表すアドレスに、スイッチ 5 0 1 から供給された 5 つの復号途中結果 D501 を格納する(書き込む)。

- 復号途中結果格納用 RAM 5 0 3 には、スイッチ 5 0 1 から 5 つの復号途中結果  
25     D412 が、復号途中結果 D502 として供給されるとともに、制御部 4 1 7 からアドレスを表す制御信号 D420<sub>3</sub> が供給される。復号途中結果格納用 RAM 5 0 3 は、制御信号 D420<sub>3</sub> が表すアドレスに既に格納されている前回の計算部 4 1 2 による第

1の演算の結果得られた5つの復号途中結果D502を読み出し、復号途中結果D504としてスイッチ504に供給する。また、復号途中結果格納用RAM502は、制御信号D420<sub>3</sub>が表すアドレスに、スイッチ501から供給された5つの復号途中結果D502を格納する(書き込む)。

- 5     スイッチ504には、復号途中結果格納用RAM502から復号途中結果D503が供給されるか、あるいは復号途中結果格納用RAM503から復号途中結果D504が供給される。また、制御部417から、復号途中結果格納用RAM502または復号途中結果格納用RAM503の一方の選択を表す制御信号D420<sub>4</sub>が供給される。スイッチ504は、制御信号D420<sub>1</sub>に基づいて、復号途中結果格納用RAM502
- 10    または復号途中結果格納用RAM503の一方を選択し、その選択した一方から供給された5つの復号途中結果を、5つの復号途中結果D413として計算部412とサイクリックシフト回路414に供給する。

- 図24は、復号途中結果格納用メモリ413の復号途中結果格納用RAM502と復号途中結果格納用RAM503の読み出しと書き込みの動作を説明するタイミ
- 15    ングチャートである。

なお、図24において、横軸は、時間(t)を表している。

- 復号途中結果格納用メモリ413では、計算部412により第1の演算が行われる場合、復号途中結果格納用RAM502が、制御信号D420<sub>2</sub>に基づいて、既に格納している、前回の計算部412の第1の演算の結果得られた復号途中結果
- 20    D501のうち、同一アドレスに格納している検査行列の第1行目から第5行目までの1に対応する復号途中結果D501を、5つ単位で9回読み出し、スイッチ504を介して、計算部412に供給する。即ち、図15の検査行列Hの行重みは、9であるため、検査行列Hの各行の1に対応する復号途中結果は9つあり、復号途中結果格納用RAM502は、第1行目から第5行目までの1に対応する5
- 25    つの復号途中結果D501を、5つ単位で9回読み出す。

次に、復号途中結果格納用RAM503は、制御信号D420<sub>3</sub>に基づいて、既に格納している、前回の計算部412による第1の演算の結果得られた復号途中結果



- D502のうち、同一アドレスに格納している検査行列の第6行目から第10行目までの1に対応する復号途中結果D502を、5つ単位で9回続けて読み出し、スイッチ504を介して、計算部412に供給する。それと同時に、復号途中結果格納用RAM502には、計算部412により、いま行われている第1の演算の結果得られる検査行列の第1行目から第5行目までの1に対応する5つの復号途中結果D412がスイッチ501を介して、復号途中結果D501として供給され、復号途中結果格納用RAM502は、その復号途中結果D501を、制御信号D420<sub>2</sub>に基づいて、既に読み出された復号途中結果D503が格納されていたアドレスに9回続けて格納する。
- 10     その後、復号途中結果格納用RAM502は、制御信号D420<sub>2</sub>に基づいて、既に格納している、前回の計算部412による第1の演算の結果得られた復号途中結果D501のうち、同一アドレスに格納している検査行列の第1.1行目から第15行目までの1に対応する復号途中結果D501を、5つ単位で9回続けて読み出し、スイッチ504を介して、計算部412に供給する。それと同時に、復号途中結果格納用RAM503には、計算部412により、いま行われている第1の演算の結果得られる検査行列の第6行目から第10行目までの1に対応する5つの復号途中結果D412がスイッチ501を介して、復号途中結果D502として供給され、復号途中結果格納用RAM503は、その復号途中結果D502を、制御信号D420<sub>3</sub>に基づいて、既に読み出された復号途中結果D504が格納されていたアドレスに9回続けて格納する。
- 20     以後、同様に、計算部412による第1の演算の結果得られる検査行列の全ての1に対応する復号途中結果が、復号途中結果格納用RAM502または復号途中結果格納用RAM503に格納されるまで、復号途中結果格納用RAM502と復号途中結果格納用RAM503は、9回ずつの読み出しまたは書き込みを交互に行う。
- 25     復号途中結果格納用メモリ413では、計算部415による第2の演算が行われる場合、制御信号D420<sub>2</sub>に基づいて、復号途中結果格納用RAM502から既に格納されている第1の演算の結果得られる復号途中結果D503を読み出すか、あ

るいは制御信号 D420<sub>3</sub>に基づいて、復号途中結果格納用 RAM 503 から、既に格納されている第 1 の演算の結果得られる復号途中結果 D504 を読み出し、その読み出した復号途中結果をスイッチ 504 を介して、サイクリックシフト回路 414 に供給する。

- 5 図 25 は、図 18 の復号装置 400 の復号処理を説明するフローチャートである。この処理は、例えば、受信用メモリ 416 に復号すべき受信データが格納されたとき、開始される。

ステップ S50 において、サイクリックシフト回路 414 は、復号途中結果格納用メモリ 413 から供給された後述するステップ S56 で格納される 5 つの復  
10 号途中結果 D413 を、サイクリックシフトし、計算部 415 に供給する。

- 具体的には、サイクリックシフト回路 414 には、復号途中結果格納用メモリ 413 から 5 つの復号途中結果 D413 が供給されるとともに、制御部 417 から、その復号途中結果 D413 に対応する検査行列の 1 が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix デー  
15 タ) を表す制御信号 D421 が供給される。サイクリックシフト回路 414 は、制御信号 D421 を元に、5 つの復号途中結果 D413 をサイクリックシフトし (並べ替え)、その結果を復号途中結果 D414 として、計算部 415 に供給する。

- なお、受信用メモリ 416 から供給された受信データ D417 に対して、まだ第 1 の演算が行われておらず、復号途中結果格納用メモリ 413 に復号途中結果  
20 D413 が格納されていない場合、計算部 415 は、復号途中結果  $u_j$  を初期値に設定する。

ステップ S51 において、計算部 415 は、第 2 の演算を行い、その演算の結果である復号途中結果 D415 を復号途中結果格納用メモリ 410 に供給する。

- 具体的には、計算部 415 には、ステップ S50 でサイクリックシフト回路 4  
25 14 から 5 つの復号途中結果 D414 が供給されるとともに、受信データ用メモリ 416 から 5 つの受信データ D417 が供給され、復号途中結果 D415 と受信データ D417 が、計算部 415 の計算器 415<sub>1</sub> 乃至 415<sub>5</sub> それぞれに 1 つずつ供給

される。さらに、計算部 4 1 5 には、制御部 4 1 7 から制御信号 D422 が供給され、その制御信号 D422 が計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> に供給される。

- 計算器 4 1 5<sub>1</sub> 乃至 4 1 5<sub>5</sub> は、復号途中結果 D414 と受信データ D417 を用いて、制御信号 D422 に基づいて、式 (5) にしたがって、それぞれ演算を行い、
- 5 その演算の結果得られる検査行列の列に対応する復号途中結果 D415(v) を復号途中結果格納用メモリ 4 1 0 に供給する。

ステップ S 5 1 の処理後は、ステップ S 5 2 に進み、復号途中結果格納用メモリ 4 1 0 は、ステップ S 5 1 で計算部 4 1 5 から供給された復号途中結果 D415 を、同一アドレスに格納し、ステップ S 5 3 に進む。

- 10 ステップ S 5 3 において、制御部 4 1 7 は、計算部 4 1 5 により、検査行列の列に対応する全ての復号途中結果 D415 が演算されたかどうかを判定し、全ての復号途中結果 D415 が演算されていないと判定した場合、ステップ S 5 0 に戻り、上述した処理を繰り返す。

- 一方、ステップ S 5 3 において、制御部 4 1 7 は、計算部 4 1 5 により、検査
- 15 行列の列に対応する全ての復号途中結果 D415 が演算されたと判定した場合、ステップ S 5 4 に進み、サイクリックシフト回路 4 1 1 は、復号途中結果格納用メモリ 4 1 0 から供給される復号途中結果 D410(v) をサイクリックシフトする。

- 具体的には、サイクリックシフト回路 4 1 1 には、復号途中結果格納用メモリ 4 1 0 から 5 つの復号途中結果 D410 が供給されるとともに、制御部 4 1 7 から、
- 20 その復号途中結果 D410 に対応する検査行列の 1 が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D418 が供給される。サイクリックシフト回路 4 1 1 は、制御信号 D418 を元に、5 つの復号途中結果 D410 をサイクリックシフトし (並べ替え)、その結果を復号途中結果 D411 として、計算部 4 1 2 に供給する。

- 25 ステップ S 5 4 の処理後は、ステップ S 5 5 に進み、計算部 4 1 2 は、第 1 の演算を行い、その演算結果である復号途中結果 D412 をサイクリックシフト回路 4 1 4 に供給する。

具体的には、計算部 4 1 2 には、ステップ S 5 4 でサイクリックシフト回路 4 1 1 から 5 つの復号途中結果 D411(v) が供給されるとともに、後述するステップ S 5 6 で既に格納された前回の計算部 4 1 2 による第 1 の演算の結果得られた 5 つの復号途中結果 D412 (D413) ( $u_j$ ) が供給され、その復号途中結果 D411 と復号途中結果 D413 が、計算部 4 1 2 の計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> のそれぞれに 1 つずつ供給される。さらに、計算部 4 1 2 には、制御部 4 1 7 から制御信号 D419 が供給され、その制御信号 D419 が計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> に供給される。

計算器 4 1 2<sub>1</sub> 乃至 4 1 2<sub>5</sub> は、それぞれ復号途中結果 D411 と復号途中結果 D413 とを用いて、制御信号 D419 に基づいて、式 (7) と式 (8) にしたがって、  
10 それぞれ演算を行い、その演算の結果得られる復号途中結果 D412 ( $u_j$ ) を復号途中結果格納用メモリ 4 1 3 に供給する。

ステップ S 5 5 の処理後は、ステップ S 5 6 に進み、復号途中結果格納用メモリ 4 1 3 は、ステップ S 5 5 で計算部 4 1 2 から供給された 5 つの復号途中結果 D412 を、同一のアドレスに格納し、ステップ S 5 7 に進む。

15 ステップ S 5 7 において、制御部 4 1 7 は、計算部 4 1 2 により、検査行列の全ての 1 に対応する復号途中結果 D412 が演算されたかどうかを判定し、全ての復号途中結果が演算されていないと判定した場合、ステップ S 5 4 に戻り、上述した処理を繰り返す。

一方、ステップ S 5 7 において、制御部 4 1 7 は、計算部 4 1 2 により、全ての  
20 の 1 に対応する復号途中結果 D412 が演算されたと判定した場合、処理を終了する。

なお、復号装置 4 0 0 は、復号回数だけ図 2 5 の復号処理を繰り返し行ない、最後の第 2 の演算の結果得られるメッセージ D415 が、最終的な復号結果として出力される。

25 上述した説明では、復号途中結果格納用メモリ 4 1 3 は、2 つのシングルポート RAM から構成にしたが、1 つの RAM に対して読み出しと書き込みが同時に起こらないようにすれば、3 つ以上の RAM から構成してもよい。例えば、RAM の物

理的なビットが足りない場合には、複数の RAM を用いて同じ制御信号を与えることで、論理的に 1 つの RAM とみなすことができる。

また、枝データ（枝に対応するメッセージ）が欠けている箇所に関しては、メモリ格納時（復号途中結果格納用メモリ 4 1 0 と 4 1 3 へのデータ格納時）には、  
5 何のメッセージも格納せず、また、演算時（計算部 4 1 2 での第 1 の演算時と計算部 4 1 5 での第 2 の演算時）にも何の演算も行わない。

図 2 6 は、本発明を適用した図 1 5 の検査行列で表現される LDPC 符号を復号する復号装置の他の一実施の形態の構成例を示すブロック図である。

図 2 6 の復号装置 6 0 0 では、図 1 6 A と C の枝データ格納メモリ 3 1 6 が、  
10 枝データ格納メモリ 3 1 6 に比べて容量の小さい復号途中結果格納用メモリ 6 1 3 になっている。

復号装置 6 0 0 は、復号途中結果格納用メモリ 6 1 0、サイクリックシフト回路 6 1 1、5 つの計算器 6 1 2<sub>1</sub>乃至計算器 6 1 2<sub>5</sub>からなる計算部 6 1 2、復号途中結果格納用メモリ 6 1 3、サイクリックシフト回路 6 1 4、5 つの計算器 6  
15 1 5<sub>1</sub>乃至計算器 6 1 5<sub>5</sub>からなる計算部 6 1 5、受信用メモリ 6 1 6、および制御部 6 1 7 から構成される。

ここで、図 2 7 乃至図 3 0 を用いて、図 2 6 の計算部 6 1 2 の計算器 6 1 2<sub>1</sub>乃至計算器 6 1 2<sub>5</sub>、および図 3 0 の計算部 6 1 5 の計算器 6 1 5<sub>1</sub>乃至計算器 6  
20 1 5<sub>5</sub>と、図 1 0 のチェックノード計算器 1 0 1 および図 1 1 のバリアブルノード計算器 1 0 3 との関係について説明する。

図 2 7 と図 2 8 は、前述の図 1 0 のチェックノード計算器 1 0 1 と図 1 1 のバリアブルノード計算器 1 0 3 とそれぞれ同一の図である。また、図 2 9 は、計算器 6 1 2<sub>k</sub> ( $k=1, 2, \dots, 5$ ) の構成例を示しており、図 3 0 は、計算器 6 1 5<sub>k</sub> ( $k=1, 2, \dots, 5$ ) の構成例を示している。

図 2 6 の復号装置 6 0 0 では、計算器 6 1 2<sub>k</sub> がチェックノード演算を行い、  
25 計算部 6 1 5<sub>k</sub> が、バリアブルノード演算をおこなうのではなく、計算器 6 1 2<sub>k</sub> がチェックノード演算の一部を、計算器 6 1 5<sub>k</sub> がチェックノード演算の他の一

部とバリエブルノード演算を行う。

即ち、図 29 の計算器 6 1 2<sub>k</sub> は、ブロック D' と E' から構成される。ブロック D' は、図 27 のチェックノード計算器 1 0 1 の一部である、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行った値を積算するブロック D と同様に構成されている。また、ブロック E' は、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の符号ビットを乗算するブロック E と同様に構成されている。

一方、図 30 の計算器 6 1 5<sub>k</sub> は、ブロック F'、G'、H' とから構成される。ブロック F' は、図 19 のチェックノード計算器 1 0 1 の他の一部である、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の符号ビットの乗算値から、求めたい枝に対応するメッセージ  $v_i$  の符号ビットを除算するとともに、検査行列の各列の全ての枝に対応するメッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行った値の積算値から、求めたい枝に対応するメッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行った値を減算した値に対して、 $\phi^{-1}$  の演算を行うブロック F と同様に構成されている。また、ブロック G' は、メッセージ  $v_i$  の絶対値に対して  $\phi$  の演算を行うブロック G と同様に構成され、ブロック H' は、図 20 のバリエブルノード計算器 1 0 3 のバリエブルノード演算を行うブロック H と同様に構成されている。

そして、図 29 の計算器 6 1 2<sub>k</sub> は、ブロック A とブロック B による演算の結果、即ち、チェックノード演算の一部を行った復号途中結果  $w$  を復号途中結果格納用メモリ 6 1 3 に供給し、図 30 の計算器 6 1 5<sub>k</sub> は、チェックノード演算の他の一部とバリエブルノード演算を行った復号途中結果  $v_i'$  を復号途中結果格納用メモリ 6 1 0 に供給する。

従って、図 26 の復号装置 6 0 0 は、計算器 6 1 2<sub>k</sub> の演算と計算器 6 1 5<sub>k</sub> の演算とを交互に行うことにより、チェックノード演算とバリエブルノード演算を行い、復号を行うことができる。

なお、図 30 の計算器 6 1 5<sub>k</sub> では、復号途中結果格納用メモリ 6 1 0 に格納

されている求めたい枝に対応する復号途中結果  $v_i'$  を用いて、ブロック C で、計算器 6 1 2<sub>k</sub> の演算の結果得られる復号途中結果  $w$  の絶対値から、求めたい枝に対応する復号途中結果  $v_i'$  を減算するとともに、復号途中結果  $w$  の符号ビットと、求めたい枝に対応する復号途中結果  $v_i'$  の符号ビットを乗算するので、図 2 7 の

5 FIFO メモリ 1 2 7 と FIFO メモリ 1 3 3 が必要ない。

次に、計算部 6 1 2 の計算器 6 1 2<sub>1</sub> 乃至計算器 6 1 2<sub>5</sub> で行われる演算と、計算部 6 1 5 の計算器 6 1 5<sub>1</sub> 乃至計算器 6 1 5<sub>5</sub> で行われる演算について、式を用いて説明する。

10 計算部 6 1 2 は、以下の式 (9) にしたがう第 1 の演算を行い、その第 1 の演算の結果である復号途中結果  $w$  を復号途中結果格納用メモリ 6 1 3 に供給して格納させる。計算部 6 1 5 は、上述した式 (1) と、以下の式 (10) と (11) にしたがう第 2 の演算を行い、その第 2 の演算の結果である復号途中結果  $v_i'$  を復号途中結果格納用メモリ 6 1 0 に供給して格納させる。

$$w = \sum_{i=1}^{d_c} |v_i'| \times \prod_{i=1}^{d_c} \text{sign}(v_i') \quad \dots (9)$$

$$15 \quad u_j = \phi^{-1}(|w| - |v_i'|) \times \text{sign}(v_i') \times \text{sign}(w) \quad \dots (10)$$

$$v_i' = \phi(|v_i|) \times \text{sign}(v_i) \quad \dots (11)$$

即ち、式 (9) にしたがう第 1 の演算の結果得られる復号途中結果  $w$  は、式 (1)、式 (10)、式 (11) にしたがう第 2 の演算の結果得られる検査行列  $H$  の  $j$  行のすべての 1 に対応するチェックノード演算の復号途中結果  $v_i'$  の絶対値  $|v_i'|$  の総和と符号ビット  $\text{sign}(v_i')$  の乗算値とを乗算したものであるので、  
20 上述した式 (7) にしたがうチェックノード演算によって求められる  $u_j$  は、式 (10) に示すように、式 (9) にしたがう第 1 の演算の結果得られる復号途中結果  $w$  の絶対値  $|w|$  から、検査行列  $H$  の  $j$  行の、各列の “1” (枝) に対応する (複数の) 復号途中結果  $v_i'$  のうち、メッセージを求めたい枝に対応する復号途中結果  $v_i'$  の絶対値  $|v_i'|$  を引いた値を用いて表すことができる。  
25

復号装置 600 では、計算部 612 による式 (9) にしたがう第 1 の演算と、計算部 615 による式 (1)、式 (10)、式 (11) にしたがう第 2 の演算とが交互に行われ、計算部 615 が、最後の第 1 の演算の結果を用いて、式 (5) にしたがう演算を行い、その演算結果を復号結果として出力することにより、

5 LDPC 符号の繰り返し復号を行う。

即ち、復号装置 600 では、計算部 612 が、計算部 615 による第 2 の演算の結果である検査行列  $H$  の  $j$  行のすべての 1 に対応する復号途中結果  $v_i'$  を用いて、第 1 の演算を行い、その演算の結果得られる検査行列の各行に対応する復号途中結果  $w$  を復号途中結果格納用メモリ 613 に格納する。従って、復号途中

10 結果格納用メモリ 613 の容量は、検査行列の “1” の数より少ない検査行列の行数と復号途中結果  $w$  の量子化ビット数とを乗算した値となる。なお、計算部 615 は、計算部 612 による第 1 の演算の結果である検査行列  $H$  の  $i$  列の各行に対応する復号途中結果  $w$  と受信値  $u_{0i}$  を用いて、第 2 の演算を行い、その演算の結果得られる検査行列の  $i$  列の 1 (枝) に対応するチェックノード演算の復

15 号途中結果  $v_i'$  を復号途中結果格納用メモリ 610 に格納する。従って、復号途中結果格納用メモリ 610 に必要な容量は、バリアブルノード演算の結果を格納する図 16 A と B の枝データ格納メモリ 311 と同様に、検査行列の 1 の数と復号途中結果  $v_i'$  の量子化ビット数とを乗算した値となる。

従って、復号装置 600 では、図 16 A と B の枝データ格納メモリ 311 に比べて、復号途中結果格納用メモリ 610 のメモリの容量を削減することができ、

20 これにより、復号装置 600 の回路規模を小さくすることができる。

以下、図 26 の復号装置 600 の各部の動作について詳細に説明する。

復号途中結果格納用メモリ 610 は、制御信号 D618 に基づいて、計算部 615 から供給される 5 つの復号途中結果 D615 をまとめて格納すると同時に、既に

25 格納してある 5 つの復号途中結果 D615 を読み出し、復号途中結果 D610 として、サイクリックシフト回路 611 と計算部 615 に供給する。即ち、復号途中結果格納用メモリ 610 は、サイクリックシフト回路 611 に供給する復号途中結果



D610 の読み出しと、計算部 6 1 5 から供給される復号途中結果 D615 の書き込みとを、同時に行う。

5      なお、復号途中結果格納用メモリ 6 1 0 には、計算部 6 1 5 の第 2 の演算により演算された検査行列の 1 (枝) に対応する復号途中結果  $v_i'$  (第 2 の復号途中結果) が格納されるので、復号途中結果格納用メモリ 6 1 0 に格納されるデータ量、即ち、復号途中結果格納用メモリ 6 1 0 に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列の 1 の数 (全枝数) との乗算値となる。

10      復号途中結果格納用メモリ 6 1 0 は、例えば、5 つの復号途中結果を同時に読み書き可能な、2 つのシングルポート RAM から構成される。復号途中結果格納用メモリ 6 1 0 には、計算部 6 1 5 から 5 つの復号途中結果 D615 が供給されるとともに、制御部 6 1 7 から復号途中結果 D615 の読み書きを制御する制御信号 D618 が供給される。

15      サイクリックシフト回路 6 1 1 には、復号途中結果格納用メモリ 6 1 0 から 5 つの復号途中結果 D610 が供給されるとともに、制御部 6 1 7 から、その復号途中結果 D610 に対応する検査行列の 1 が、検査行列において元となる単位行列などを幾つサイクリックシフトであるかの情報 (Matrix データ) を表す制御信号 D619 が供給される。サイクリックシフト回路 6 1 1 は、制御信号 D619 を元に、5 つの復号結果 D610 を並べ替えるサイクリックシフトを行い、その結果を復号途中結果 D611 として、計算部 6 1 2 に供給する。

20      計算部 6 1 2 は、5 つの計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> からなる。計算部 6 1 2 には、サイクリックシフト回路 6 1 1 から 5 つの復号途中結果 D611 (第 2 の復号途中結果) ( $v_i'$ ) が供給され、その 5 つの復号途中結果 D611 (第 1 の復号途中結果) ( $w$ ) が、計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> のそれぞれに供給される。また、計算部 6 1 2 には、制御部 6 1 7 から制御信号 D620 が供給され、その制御信号  
25      D620 が、計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> に供給される。なお、制御信号 D620 は、5 つの計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> に共通の信号である。

計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> は、それぞれ復号途中結果 D611 を用いて、式

(9) にしたがって第1の演算を行い、復号途中結果 D612(w) を求める。計算部 612 は、計算器 612<sub>1</sub> 乃至 612<sub>5</sub> による演算の結果得られる5つの復号途中結果 D612 を復号途中結果格納用メモリ 613 に供給する。

5 復号途中結果格納用メモリ 613 には、計算部 612 から、計算部 612 による第1の演算の結果である検査行列の行に対応する5つの復号途中結果 D612 が供給され、復号途中結果格納用メモリ 613 は、計算部 612 から供給された5つの復号途中結果 D612 を、第1アドレスから順に格納（記憶）する。

10 即ち、復号途中結果格納用メモリ 613 の第1アドレスには、検査行列の行に対応する復号途中結果のうち、第1行目から第5行目の復号途中結果 w が格納される。そして、同様に、第2アドレスには、第6行目から第10行目の復号途中結果 w が格納され、第3アドレスには、第11行目から第15行目の復号途中結果 w が格納される。以後、同様に、第16行目から第30行目までの復号途中結果 w が、5個ずつ、第4アドレスから第6アドレスまで格納され、計60個の復号途中結果 w が復号途中結果格納用メモリ 613 に格納される。従って、復号途中結果格納用メモリ 610 のワード（word）数は、図15の検査行列 H の行数である30を、同時に読み書きする復号途中結果の数である5で割り算した6となる。

20 また、復号途中結果格納用メモリ 613 は、既に格納してある5つの復号途中結果 D613 から、計算部 615 が求めようとする復号途中結果  $v_i'$  の対応する検査行列 H の列において“1”になっている復号途中結果 w を5つ同時に読み出し、復号途中結果 D613 として、サイクリックシフト回路 614 に供給する。

25 なお、復号途中結果格納用メモリ 613 は、例えば、5つの復号途中結果を同時に読み書き可能な、シングルポート RAM で構成される。また、復号途中結果格納用メモリ 613 には、計算部 612 の第1の演算により演算された行に対応する復号途中結果 w が格納されるので、復号途中結果格納用メモリ 613 に格納されるデータ量、即ち、復号途中結果格納用メモリ 613 に必要とされる記憶容量は、復号途中結果の量子化ビット数と、検査行列 H の行数との乗算値であ

る。

サイクリックシフト回路 6 1 4 には、復号途中結果格納用メモリ 6 1 3 から 5 つの復号途中結果 D613 (復号途中結果  $w$ ) が供給されるとともに、制御部 6 1 7 から、その復号途中結果 D613 に対応する検査行列の 1 が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報

(Matrix データ) を表す制御信号 D621 が供給される。サイクリックシフト回路 6 1 4 は、制御信号 D621 を元に、5 つの復号途中結果 D613 を並べ替えるサイクリックシフトを行い、その結果を復号途中結果 D614 として、計算部 6 1 5 に供給する。

- 10 計算部 6 1 5 は、5 つの計算器 6 1 5<sub>1</sub> 乃至 6 1 5<sub>5</sub> からなる。バリアブルノード計算部 6 1 5 には、サイクリックシフト回路 6 1 4 から 5 つの復号途中結果 D614( $w$ ) が供給されるとともに、復号途中結果格納用メモリ 6 1 0 から 5 つの復号途中結果 D610 ( $v_i'$ ) が供給され、その復号途中結果 D614 と復号途中結果 D610 が、計算器 6 1 5<sub>1</sub> 乃至 6 1 5<sub>5</sub> のそれぞれに供給される。また、計算部 6 1 5 には、受信用メモリ 6 1 7 から 5 つの受信データ D617 が供給され、その受信データ D617 が、計算器 6 1 5<sub>1</sub> 乃至 6 1 5<sub>5</sub> のそれぞれに供給される。さらに、計算部 6 1 7 には、制御部 6 1 7 から制御信号 D622 が供給され、その制御信号 D622 が計算器 6 1 5<sub>1</sub> 乃至 6 1 5<sub>5</sub> に供給される。なお、制御信号 D622 は、5 つの計算器 6 1 7<sub>1</sub> 乃至 6 1 7<sub>5</sub> に共通の信号である。

- 20 計算器 6 1 5<sub>1</sub> 乃至 6 1 5<sub>5</sub> は、それぞれ復号途中結果 D614 と D611、受信データ D617 (LDPC 符号) とを用いて、式 (1)、式 (10)、式 (11) にしたがって、それぞれ第 2 の演算を行い、検査行列の各列の 1 に対応する 5 つの復号途中結果 D615 ( $v_i'$ ) を求める。計算部 6 1 5 は、計算器 6 1 5<sub>1</sub> 乃至 6 1 5<sub>5</sub> の第 2 の演算の結果得られる 5 つの復号途中結果 D615 を、復号途中結果格納用メモリ 6 1 0 に供給する。

受信用メモリ 6 1 6 は、通信路を通して受信した受信値 (符号ビット) D616 から計算した符号ビットの 0 らしさの値である受信 LLR (対数尤度比) を、受信

データ D617 として格納する。

即ち、受信用メモリ 616 の第 1 のアドレスには、検査行列の列に対応する受信データ D617 のうち、検査行列の第 1 列目から第 5 列目までに対応する受信データ D617 が格納される。そして、第 2 のアドレスには、検査行列の第 6 列目から第 10 列目までに対応する受信データ D617 が格納され、第 3 アドレスには、検査行列の第 11 列目から第 16 列目までに対応する受信データ D617 が格納される。以後、同様に、第 4 アドレスから第 18 アドレスまでに、検査行列の第 17 列目から第 90 列目までに対応する受信データ D617 が、5 つずつ格納される。

そして、受信用メモリ 616 は、既に格納している受信データ D617 を計算部 615 による第 2 の演算に必要となる順番に 5 つずつ同時に読み出し、計算部 615 に供給する。

なお、受信用メモリ 616 は、例えば、シングルポート RAM から構成される。また、受信用メモリ 616 に格納されるデータ量、即ち、受信用メモリ 616 に必要とされる記憶容量は、LDPC 符号の符号長と、受信データの量子化ビット数との乗算値である。さらに、受信用メモリ 616 のワード (word) 数は、LDPC 符号の符号長、即ち、検査行列の列数である 90 を、同時に読み出す受信データ D617 の数である 5 で割り算した値の 18 である。

制御部 617 は、制御信号 D618 を復号途中結果格納用メモリ 610 に、制御信号 D619 をサイクリックシフト回路 611 に供給することにより、それぞれを制御する。また、制御部 617 は、制御信号 D620 を計算部 612 に、制御信号 D621 をサイクリックシフト回路 614 に、制御信号 D622 を計算部 615 に供給することにより、それぞれを制御する。

復号途中結果格納用メモリ 610、サイクリックシフト回路 611、計算部 612、復号途中結果格納用メモリ 613、サイクリックシフト回路 614、計算部 615 の順で、データが一巡することで、復号装置 600 は、1 回の復号を行うことができる。復号装置 600 では、所定の回数だけ繰り返して復号が行われた後、計算部 615 が、式 (5) にしたがう演算を行い、その演算結果が最終的

な復号結果として出力される。

図 29 は、図 26 の計算部 612 の計算器 612<sub>1</sub> の構成例を示すブロック図である。

5      なお、図 29 では、計算器 612<sub>1</sub> について説明するが、計算器 612<sub>2</sub> 乃至計算器 612<sub>5</sub> も同様に構成される。

10      また、図 29 では、計算器 615 による第 2 の演算の結果得られる各復号途中結果 ( $v_i'$ ) が 6 ビットに量子化されているものとして、計算器 612<sub>1</sub> を表している。さらに、図 29 の計算器 612<sub>1</sub> には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

図 29 の計算器 612<sub>1</sub> は、制御部 617 から供給される制御信号 D620 に基づいて、サイクリックシフト回路 611 から 1 つずつ読み込まれる復号途中結果 D611 ( $v_i'$ ) を用いて、式 (9) にしたがう第 1 の演算を行う。

15      即ち、計算器 612<sub>1</sub> には、サイクリックシフト回路 611 から供給される 5 つの 6 ビットの復号途中結果 D611 ( $v_i'$ ) のうちの、1 つの復号途中結果 D611 が供給され、最上位ビットの符号ビット D631 が EXOR 回路 635 に供給されるとともに、その 6 ビットの復号途中結果 D611 ( $v_i'$ ) の下位 5 ビットの絶対値 D632 ( $|v_i'|$ ) が、加算器 631 に供給される。また、計算器 612<sub>1</sub> には、制御部 617 から制御信号 D620 が供給され、その制御信号 D620 がセクタ 633  
20      とセクタ 637 に供給される。

加算器 631 は、絶対値 D632 ( $|v_i'|$ ) とレジスタ 632 に格納されている 9 ビットの値 D633 とを加算することにより、絶対値 D632 ( $|v_i'|$ ) を積算し、その結果得られる 9 ビットの積算値をレジスタ 632 に再格納する。なお、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D611 から求められた絶対値  
25      D632 ( $|v_i'|$ ) が積算された場合、レジスタ 632 はリセットされる。

検査行列の 1 行に亘る復号途中結果 D611 が 1 つずつ読み込まれ、レジスタ 632 に 1 行分の絶対値 D632 が積算された積算値が格納された場合、制御部 61

7 から供給される制御信号 D620 は、0 から 1 に変化する。例えば、行の重み (row weight) が「9」である場合、制御信号 D620 は、1 から 8 クロック目までは、「0」となり、9 クロック目では「1」となる。

制御信号 D620 が「1」の場合、セクタ 6 3 3 は、レジスタ 6 3 2 に格納さ  
5 れている値、即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D611 (復号途中結果  $v_i'$ ) の絶対値 D632 ( $|v_i'|$ ) が積算された 9 ビットの値 D633 ( $i = 1$  から  $i = d$  までの  $\sum |v_i'|$ ) を選択し、値 D634 として、レジスタ 6 3 4 に出力して格納させる。レジスタ 6 3 4 は、格納している値 D634 を、9 ビットの値 D635 として、セクタ 6 3 3 に供給するとともに、出力する。制御  
10 信号 D620 が「0」の場合、セクタ 6 3 3 は、レジスタ 6 3 4 から供給された値 D635 を選択し、レジスタ 6 3 4 に出力して再格納させる。即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D611 (復号途中結果  $v_i'$ ) の絶対値 D632 ( $|v_i'|$ ) が積算されるまで、レジスタ 6 3 4 は、前回積算された  $|v_i'|$  を、セクタ 6 3 3 に供給するとともに、出力する。

15 以上の処理と並行して、EXOR 回路 6 3 5 は、レジスタ 6 3 6 に格納されている 1 ビットの値 D637 と符号ビット D631 との排他的論理和を演算することにより、符号ビットどうしの乗算を行い、1 ビットの乗算結果 D636 をレジスタ 6 3 6 に再格納する。なお、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D611 の符号ビット D631 が乗算された場合、レジスタ 6 3 6 はリセットされ  
20 る。

検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D611 から求められた符号ビット D631 が乗算された乗算結果 D636 ( $i = 1$  から  $d$  までの  $\prod \text{sign}(v_i')$ ) がレジスタ 6 3 6 に格納された場合、制御部 6 1 7 から供給される制御信号 D620 は、「0」から「1」に変化する。

25 制御信号 D620 が「1」の場合、セクタ 6 3 7 は、レジスタ 6 3 6 に格納されている値、即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D611 の符号ビット D631 が乗算された値 D637 ( $i = 1$  から  $i = d$  までの

$\Pi \text{sign}(v_i')$  を選択し、1ビットの値 D638 としてレジスタ 638 に出力して格納させる。レジスタ 638 は、格納している値 D638 を、1ビットの値 D639 としてセレクタ 637 に供給するとともに、出力する。制御信号 D620 が「0」の場合、セレクタ 637 は、レジスタ 638 から供給された値 D639 を選択し、  
5 レジスタ 638 に出力して再格納させる。即ち、検査行列の 1 行に亘る全ての 1 に対応する復号途中結果 D611 (復号途中結果  $v_i'$ ) の符号ビット D631 が乗算されるまで、レジスタ 638 は、前回格納した値を、セレクタ 637 に供給するとともに、出力する。

10 計算器 612<sub>i</sub> では、レジスタ 634 から出力された 9 ビットの値 D635 ( $i = 1$  から  $i = d_0$  までの  $\sum |v_i'|$ ) を下位 9 ビットとするとともに、レジスタ 638 から出力された 1 ビットの値 D639 ( $\text{sign}(v_i')$ ) を最上位ビットとする合計 10 ビットが復号途中結果 D612 (復号途中結果  $w$ ) として出力される。

以上のように、計算器 612<sub>i</sub> では、式 (9) の演算が行われ、復号途中結果  $w$  が求められる。

15 図 30 は、図 26 の計算部 615 の計算器 615<sub>1</sub> の構成例を示すブロック図である。

なお、図 30 では、計算器 615<sub>1</sub> について説明するが、計算器 615<sub>2</sub> 乃至計算器 615<sub>g</sub> も同様に構成される。

また、図 30 では、計算器 612 による第 1 の演算の結果得られる各復号途中  
20 結果 ( $w$ ) が符号ビットを合わせて合計 10 ビットに量子化され、復号途中結果格納用メモリ 610 から供給される、前回の第 2 の演算の結果得られた各復号途中結果 ( $u_j$ ) が符号ビットを合わせて 6 ビットに量子化されているものとして、計算器 615<sub>1</sub> を表している。さらに、図 30 の計算器 615<sub>1</sub> には、クロック  $ck$  が供給され、このクロック  $ck$  は、必要なブロックに供給されるようになっている。  
25 る。そして、各ブロックは、クロック  $ck$  に同期して処理を行う。

図 30 の計算器 615<sub>1</sub> は、制御部 617 から供給される制御信号 D622 に基づいて、受信用メモリ 616 から 1 つずつ読み込まれる受信データ D617 (受信

値  $u_{0i}$  )、サイクリックシフト回路 6 1 4 から 1 つずつ読み込まれる復号途中結果 D614 (w)、および復号途中結果格納用メモリ 6 1 0 から 1 つずつ読み込まれる前回の計算部 6 1 5 による第 2 の演算の結果得られた復号途中結果 D610

( $v_i'$ ) とを用いて、式 (1)、式 (10)、式 (11) にしたがう第 2 の演算

5 を行う。

即ち、計算器 6 1 5<sub>1</sub> では、サイクリックシフト回路 6 1 4 から、検査行列の行に対応する 10 ビットの復号途中結果 D614 (復号途中結果 w) が 1 つずつ読み込まれとともに、復号途中結果格納用メモリ 6 1 0 から、前回の計算部 6 1 5 による第 2 の演算の結果得られた 6 ビットの復号途中結果 D610 (復号途中結果  $v_i'$ ) が 1 つずつ読み込まれ、その復号途中結果 D614 の最上位ビットの符号ビット D651 ( $\text{sign}(w)$ ) と復号途中結果 D610 の最上位ビットの符号ビット D653

( $\text{sign}(u_j)$ ) が、EXOR 回路 6 5 3 に供給されるとともに、その復号途中結果 D614 の下位 9 ビットの絶対値 D652 ( $|w|$ ) と復号途中結果 D610 の下位 9 ビットの符号ビット D653 ( $|v_i'|$ ) が、減算器 6 5 1 に供給される。また、計算器 6

15 1 5<sub>1</sub> では、受信用メモリ 6 1 6 から 6 ビットの受信データ D617 が 1 つずつ読み込まれ、加算器 6 5 8 に供給される。さらに、計算器 6 1 5<sub>1</sub> には、制御部 6 1 7 から制御信号 D622 が供給され、その制御信号 D622 は、セレクタ 6 5 6 に供給される。

減算器 6 5 1 は、絶対値 D652 から絶対値 D654 を減算し、その 5 ビットの減算値 D655 を LUT 6 5 2 に供給する。LUT 6 5 2 は、その減算値 D655 に対して、 $\phi^{-1}$  の演算を行った 5 ビットの演算結果 D656 ( $\phi^{-1}(|w| - |v_i'|)$ ) を出力する。

一方、EXOR 回路 6 5 3 は、符号ビット D651 ( $\text{sign}(w)$ ) と符号ビット D653 ( $\text{sign}(v_i')$ ) との排他的論理和を演算することにより、符号ビット D651 と

25 符号ビット D653 を乗算し、1 ビットの乗結果を乗算値 D657 として出力する。そして、LUT 6 5 2 から供給される 5 ビットの演算結果 D656 を下位 5 ビット ( $\phi^{-1}(|w| - |v_i'|)$ ) とするとともに、EXOR 回路 6 5 3 から供給される 1 ビ



ットの値 D657 ( $\text{sign}(w) \times \text{sign}(v_i')$ ) を最上位ビットとした 6 ビットの値 D658 が、加算器 6 5 4 に供給されるとともに、FIFO メモリ 6 5 9 に供給される。

以上のように、式 (10) にしたがう演算が行われ、その演算の結果である 6 ビットの値 D658 ( $u_j$ ) が、加算器 6 5 4 に供給されるとともに、FIFO メモリ 6 5 9 に供給される。

加算器 6 5 4 は、6 ビットの値 D658 ( $u_j$ ) とレジスタ 6 5 5 に格納されている 9 ビットの値 D659 とを加算することにより、値 D658 を積算し、その結果得られる 9 ビットの積算値を、レジスタ 6 5 5 に再格納する。なお、検査行列の 1 列に亘る全ての 1 に対応する値 D658 が積算された場合、レジスタ 6 5 5 はリセットされる。

検査行列の 1 列に亘る値 D658 が 1 つずつ読み込まれ、レジスタ 6 5 5 に 1 列分の値 D658 が積算された値が格納された場合、制御部 6 1 7 から供給される制御信号 D622 は、「0」から「1」に変化する。例えば、列の重みが「5」である場合、制御信号 D622 は、1 から 4 クロック目までは「0」となり、5 クロック目では「1」となる。

制御信号 D622 が「1」の場合、セクタ 6 5 6 は、レジスタ 6 5 5 に格納されている値、即ち、検査行列の 1 列に亘る 1 に対応する値 D658 ( $u_j$ ) が積算された 9 ビットの値 D659 ( $j = 1$  から  $d_v$  までの  $\sum u_j$ ) を選択し、レジスタ 6 5 7 に出力して格納させる。レジスタ 6 5 7 は、格納している値 D659 を、9 ビットの値 D660 として、セクタ 4 7 1 と加算器 6 5 8 に供給する。制御信号 D622 が「0」の場合、セクタ 6 5 6 は、レジスタ 6 5 7 から供給された値 D660 を選択し、レジスタ 6 5 7 に出力し再格納させる。即ち、検査行列の 1 列に亘る 1 に対応する値 D658 ( $u_j$ ) が積算されるまで、レジスタ 6 5 7 は、前回積算された値を、セクタ 6 5 6 と加算器 6 5 8 に供給する。

加算器 6 5 8 は、9 ビットの値 D660 と、受信用メモリ 6 1 6 から供給された 6 ビットの受信データ D617 とを加算して、その結果得られる 9 ビットの値 D661 を供給する。

計算器 6 1 5 では、最後の演算を行う場合、加算器 6 5 8 が、9 ビットの値 D661 を最終的な復号結果として出力する。即ち、計算部 6 1 5 は、式 (5) にしたがって演算を行う。

一方、FIFO メモリ 6 5 9 は、レジスタ 6 6 5 から新たな値 D660 ( $j = 1$  から  $j = d_v$  までの  $\sum u_j$ ) が出力されるまでの間、6 ビットの値 D658 ( $u_j$ ) を遅延し、6 ビットの値 D662 として減算器 6 6 0 に供給する。減算器 6 6 0 は、9 ビットの値 D660 から 6 ビットの値 D662 を減算し、その減算値 D663 を出力する。即ち、減算器 6 6 0 は、検査行列の 1 列に亘る 1 に対応する値 D658 の積算値から、求めたい枝に対応する値、即ち検査行列の所定の 1 に対応する値 D658 ( $u_j$ ) を減算して、その減算値 ( $i = 1$  から  $i = d_v - 1$  までの  $\sum u_j$ ) を 6 ビットの減算値 D663 として出力する。

以上のように、式 (1) にしたがう演算が行われ、その演算の結果である 6 ビットの減算値 D663 ( $v_i$ ) が出力される。そして、減算器 6 6 0 から出力された 6 ビットの減算値 D663 の下位 5 ビットの絶対値 ( $|v_i|$ ) が、LUT 6 6 1 に供給されるとともに、最上位ビットの符号ビット ( $\text{sign}(v_i)$ ) が値 D665 として出力される。

LUT 6 6 1 は、絶対値 ( $|v_i|$ ) に対して、 $\phi$  の演算を行った 5 ビットの演算結果 D666 ( $\phi(|v_i|)$ ) を出力する。そして、LUT 6 6 1 から出力された 5 ビットの演算結果 D666 ( $\phi(|v_i|)$ ) を下位 5 ビットとするとともに、値 D665 ( $\text{sign}(v_i)$ ) を最上位ビットとした合計 6 ビットを、復号途中結果 ( $v_i'$ ) として復号途中結果格納用メモリ 6 1 0 に供給する。

以上のように、計算器 6 1 5<sub>1</sub> では、式 (1)、式 (10)、式 (11) の演算が行われ、復号途中結果  $v_i'$  が求められる。

なお、図 15 の検査行列の列の重みの最大は 5 であるため、即ち、計算器 6 1 5<sub>1</sub> に供給される復号途中結果 D614 ( $w$ ) と復号途中結果 D610 ( $v_i'$ ) の最大数は 5 であるため、計算器 6 1 5<sub>1</sub> は、5 個の復号途中結果 D614 と復号途中結果 D610 から求められる 5 個の演算結果 D658 ( $u_j$ ) を遅延させる FIFO メモリ 6 5

9を有している。列の重みが5未満の行のメッセージを計算するときには、FIFOメモリ659における遅延量が、その列の重みの値に減らされる。

図31は、図26の復号途中結果格納用メモリ610の構成例を示すブロック図である。

- 5 復号途中結果格納用メモリ610は、スイッチ701と704、および2つのシングルポートRAMである復号途中結果格納用RAM702と703から構成される。

この復号途中結果格納用メモリ610の各部について詳細に説明する前に、まず、復号途中結果格納用RAM702と703へのデータの格納方法について説明  
10 する。

復号途中結果格納用RAM702と703は、計算部612による第1の演算の結果得られ、スイッチ701を介して供給された復号途中結果D615を格納する。

具体的には、復号途中結果格納用RAM702の第1アドレスから第5アドレス  
15 には、図15の検査行列Hの第1列目から第5列目までの1に対応する復号途中結果D615(D701)が、各行ともに横方向(列方向)に詰めた形に(0を無視した形で)格納される。

即ち、第j行第i列を、 $(j, i)$ と表すこととすると、復号途中結果格納用RAM702の第1アドレスには、図15の検査行列の $(1, 1)$ から $(5, 5)$ の $5 \times 5$ の単位行列の1に対応するデータが、第2アドレスには、図15の検査行列の $(6, 1)$   
20 から $(10, 5)$ の和行列( $5 \times 5$ の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2のシフト行列との和である和行列)を構成する第1のシフト行列の1の位置に対応するデータが格納される。また、第3アドレスには、検査行列の $(6, 1)$ から $(10, 5)$ の和行列を構成する第2のシフト行列の1の位置に対応するデータが格納される。以下、  
25 第4アドレスおよび第5アドレスについても、図15の検査行列に対応づけて、データが格納される。

復号途中格納用RAM702の第6アドレスから第10アドレスには、図15の

検査行列の第 1 1 列目から第 1 5 列目までの 1 に対応するデータが格納される。即ち、第 6 アドレスには、検査行列の (11, 11) から (15, 15) の和行列 (5 × 5 の単位行列と、5 × 5 の単位行列を右に 3 つだけサイクリックシフトした第 1 のシフト行列との和である和行列) を構成する第 1 のシフト行列の 1 の位置に対応するデータが格納され、第 7 アドレスには、検査行列の (11, 11) から (15, 15) の和行列を構成する単位行列の 1 に対応するデータが格納される。以下、第 8 アドレスから第 1 0 アドレスについても、検査行列に対応づけてデータが格納される。

同様に、復号途中格納用 RAM 7 0 2 の第 1 0 アドレスから第 2 8 アドレスには、図 1 5 の検査行列に対応づけて、第 2 1 列目から第 2 5 列目まで、第 3 1 列目から第 3 5 列目まで、第 4 1 列目から第 4 5 列目まで、第 5 1 列目から第 5 5 列目まで、第 6 1 列目から第 6 5 列目まで、第 7 1 列目から第 7 5 列目まで、第 8 1 列目から第 8 5 列目までの 1 に対応するデータが格納される。即ち、復号途中結果格納用 RAM 7 0 2 のワード数は、2 8 である。

復号途中結果格納用 RAM 7 0 3 の第 1 アドレスから第 5 アドレスには、図 1 5 の検査行列 H の第 6 列目から第 1 0 列目までの 1 に対応する復号途中結果 D615 (D702) が、各行ともに横方向 (列方向) に詰めた形に (0 を無視した形で) 格納される。

即ち、復号途中結果格納用 RAM 7 0 3 の第 1 アドレスには、検査行列の構成行列である (6, 1) から (10, 5) の和行列 (5 × 5 の単位行列を右に 1 つだけサイクリックシフトした第 1 のシフト行列と、右に 2 つだけサイクリックシフトした第 2 のシフト行列の和である和行列) を構成する第 1 のシフト行列の 1 に対応するデータが、第 2 アドレスには、検査行列の構成行列である (6, 1) から (10, 5) の和行列を構成する第 2 のシフト行列の 1 に対応するデータが格納される。以下、第 3 アドレスから第 5 アドレスも同様に検査行列の構成行列と対応づけてデータが格納される。

同様に、復号途中格納用 RAM 7 0 3 の第 6 アドレスから第 2 6 アドレスには、図 1 5 の検査行列の第 1 6 列目から第 2 0 列目まで、第 2 6 列目から第 3 0 列目

まで、第36列目から第40列目まで、第46列目から第50列目まで、第56列目から第60列目まで、第66列目から第70列目まで、第76列目から第80列目まで、第86列目から第90列目までの1に対応するデータが、図15の検査行列に対応づけて格納される。即ち、復号途中結果格納用RAM703のワード数は、26である。

上述したように、復号途中結果格納用RAM702のワード(word)数は、28であり、復号途中結果格納用RAM703のワード数は、26である。

図32は、復号途中結果格納用メモリ610の復号途中結果格納用RAM702と復号途中結果格納用RAM703の読み出しと書き込みの動作を説明するタイミングチャートである。

なお、図32において、横軸は、時間(t)を表している。

復号途中結果格納用メモリ610では、計算部612による第1の演算が行われる場合、制御部617から供給される制御信号D720<sub>2</sub>に基づいて、復号途中結果格納用RAM702から既に格納されている第2の演算の結果得られる復号途中結果D703を読み出すか、あるいは制御部617から供給される制御信号D720<sub>3</sub>に基づいて、復号途中結果格納用RAM703から、既に格納されている第2の演算の結果得られる復号途中結果D704を読み出し、その読み出した復号途中結果をスイッチ704を介して、サイクリックシフト回路614に供給する。

復号途中結果格納用メモリ610には、計算部615により第2の演算が行われる場合、計算部615から第2の演算の結果得られる復号途中結果D615

( $v_i'$ )が供給され、その復号途中結果D615が復号途中結果格納用RAM702または復号途中結果格納用RAM703のうちの一方の所定のアドレスに書き込まれると同時に、他方から、前回の計算部615による第2の演算の結果得られた復号途中結果D610( $v_i'$ )が読み出され、サイクリックシフト回路614を介して、計算部615に出力される。

スイッチ701には、計算部615から5つの復号途中結果D615が供給されるとともに、その復号途中結果D615を書き込むメモリとして、復号途中結果格

納用 RAM 7 0 2 または復号途中結果格納用 RAM 7 0 3 の一方の選択を表す制御信号  $D720_1$  が供給される。スイッチ 7 0 1 は、制御信号  $D720_1$  に基づいて、復号途中結果格納用 RAM 7 0 2 または復号途中結果格納用 RAM 7 0 3 の一方を選択し、その選択した一方に、5つの復号途中結果  $D612$  を供給する。

- 5 復号途中結果格納用 RAM 7 0 2 には、スイッチ 7 0 1 から5つの復号途中結果  $D612$  が、復号途中結果  $D701$  として供給されるとともに、制御部 6 1 7 からアドレスを表す制御信号  $D720_2$  が供給される。復号途中結果格納用 RAM 7 0 2 は、制御信号  $D720_2$  が表すアドレスに既に格納されている前回の計算部 6 1 5 による第2の演算の結果得られた5つの復号途中結果  $D701$  を読み出し、復号途中結果
- 10  $D703$  としてスイッチ 7 0 4 に供給する。また、復号途中結果格納用 RAM 7 0 2 は、制御信号  $D720_2$  が表すアドレスに、スイッチ 7 0 1 から供給された5つの復号途中結果  $D702$  を格納する(書き込む)。

- 復号途中結果格納用 RAM 7 0 3 には、スイッチ 7 0 1 から5つの復号途中結果  $D615$  が、復号途中結果  $D702$  として供給されるとともに、制御部 6 1 7 からアドレスを表す制御信号  $D720_3$  が供給される。復号途中結果格納用 RAM 7 0 3 は、制御信号  $D720_3$  が表すアドレスに既に格納されている前回の計算部 6 1 5 による第2の演算の結果得られた5つの復号途中結果  $D702$  を読み出し、復号途中結果
- 15  $D704$  としてスイッチ 7 0 4 に供給する。また、復号途中結果格納用 RAM 7 0 2 は、制御信号  $D720_3$  が表すアドレスに、スイッチ 7 0 1 から供給された5つの復号途中結果  $D702$  を格納する(書き込む)。
- 20

- スイッチ 7 0 4 には、復号途中結果格納用 RAM 7 0 2 から復号途中結果  $D703$  が供給されるか、あるいは復号途中結果格納用 RAM 7 0 3 から復号途中結果  $D704$  が供給される。また、制御部 6 1 7 から、復号途中結果格納用 RAM 7 0 2 または復号途中結果格納用 RAM 7 0 3 の一方の選択を表す制御信号  $D720_4$  が供給
- 25 される。スイッチ 7 0 4 は、制御信号  $D720_4$  に基づいて、復号途中結果格納用 RAM 7 0 2 または復号途中結果格納用 RAM 7 0 3 の一方を選択し、その選択した一方から供給された5つの復号途中結果を、5つの復号途中結果  $D610$  として計

算部 6 1 5 に供給する。

復号途中結果格納用メモリ 6 1 0 では、計算部 6 1 5 により第 2 の演算が行われる場合、復号途中結果格納用 RAM 7 0 2 が、制御信号  $D720_2$  に基づいて、既に格納している、前回の計算部 6 1 5 の第 2 の演算の結果得られた復号途中結果 D701 のうち、同一アドレスに格納している検査行列の第 1 列目から第 5 列目までの 1 に対応する復号途中結果 D701 を、5 つ単位で、5 回読み出し、スイッチ 7 0 4 を介して、計算部 6 1 5 に供給する。即ち、図 1 5 の検査行列 H の列重みは、5 であるため、検査行列 H の各列の 1 に対応する復号途中結果は 5 つあり、復号途中結果格納用 RAM 7 0 2 は、第 1 行列目から第 5 列目までの 1 に対応する復号途中結果 D701 を、5 つ単位で 5 回読み出す。

次に、復号途中結果格納用 RAM 7 0 3 は、制御信号  $D720_3$  に基づいて、既に格納している、前回の計算部 6 1 5 による第 2 の演算の結果得られた復号途中結果 D702 のうち、同一アドレスに格納している検査行列の第 6 列目から第 1 0 列目までの 1 に対応する 5 つの復号途中結果 D702 を、5 回続けて読み出し、スイッチ 7 0 4 とサイクリックシフト回路 6 1 4 を介して、計算部 6 1 5 に供給する。それと同時に、復号途中結果格納用 RAM 7 0 2 には、計算部 6 1 5 により、いま行われている第 2 の演算の結果得られる検査行列の第 1 列目から第 5 列目までの 1 に対応する 5 つの復号途中結果 D615 がスイッチ 7 0 1 を介して、復号途中結果 D701 として供給され、復号途中結果格納用 RAM 7 0 2 は、その復号途中結果 D701 を、制御信号  $D720_2$  に基づいて、既に読み出された復号途中結果 D703 が格納されていたアドレスに 5 回続けて格納する。

その後、復号途中結果格納用 RAM 7 0 2 は、制御信号  $D720_2$  に基づいて、既に格納している、前回の計算部 6 1 5 による第 2 の演算の結果得られた復号途中結果 D701 のうち、同一アドレスに格納している検査行列の第 1 1 列目から第 1 5 列目までの 1 に対応する復号途中結果 D701 を、5 つ単位で 5 回続けて読み出し、スイッチ 7 0 4 を介して、計算部 6 1 5 に供給する。それと同時に、復号途中結果格納用 RAM 7 0 3 には、計算部 6 1 5 により、いま行われている第 2 の演算の

結果得られる検査行列の第 6 列目から第 10 列目までの 1 に対応する 5 つの復号途中結果 D612 がスイッチ 701 を介して、復号途中結果 D702 として供給され、復号途中結果格納用 RAM 703 は、その復号途中結果 D702 を、制御信号 D720、に基づいて、既に読み出された復号途中結果 D704 が格納されていたアドレスに 5 5 回続けて格納する。

以後、同様に、計算部 615 による第 2 の演算の結果得られる検査行列の全ての 1 に対応する復号途中結果が、復号途中結果格納用 RAM 702 または復号途中結果格納用 RAM 703 に格納されるまで、復号途中結果格納用 RAM 702 と復号途中結果格納用 RAM 703 は、5 回ずつの読み出しまたは書き込みを交互に行う。

10 図 33 は、図 26 の復号装置 600 の復号処理を説明するフローチャートである。この処理は、例えば、受信用メモリ 616 に復号すべき受信データが格納されたとき、開始される。

ステップ S70 において、サイクリックシフト回路 614 は、復号途中結果格納用メモリ 613 から供給された後述するステップ S76 で格納された 5 つの復号途中結果 D613 を、並べ替えてサイクリックシフトを行い、計算部 615 に供給する。 15

具体的には、サイクリックシフト回路 614 には、復号途中結果格納用メモリ 613 から 5 つの復号途中結果 D613 が供給されるとともに、制御部 617 から、その復号途中結果 D613 に対応する検査行列の 1 が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D621 が供給される。サイクリックシフト回路 614 は、制御信号 D621 を元に、5 つの復号途中結果 D613 をサイクリックシフトし (並べ替え)、その結果を復号途中結果 D614 として、計算部 615 に供給する。 20

なお、受信用メモリ 616 から供給された受信データ D617 に対して、まだ第 1 の演算が行われておらず、復号途中結果格納用メモリ 613 に復号途中結果 D612 が格納されていない場合、計算部 615 は、初期値に設定する。 25

ステップ S71 において、計算部 615 は、第 2 の演算を行い、その演算の結



果である復号途中結果 D615 を復号途中結果格納用メモリ 610 に供給する。

具体的には、計算部 615 には、ステップ S70 でサイクリックシフト回路 614 から 5 つの復号途中結果 D614 が供給されるとともに、直前の後述するステップ S72 で復号途中結果格納用メモリ 610 から復号途中結果 D610 が供給される。また、受信データ用メモリ 616 から 5 つの受信データ D617 が供給され、5 つの復号途中結果 D615 と D610、受信データ D617 が、計算部 615 の計算器 615<sub>1</sub>乃至 615<sub>5</sub>のそれぞれに 1 つずつ供給される。さらに、計算部 615 には、制御部 617 から制御信号 D622 が供給され、その制御信号 D622 が計算器 615<sub>1</sub>乃至 615<sub>5</sub>に供給される。

10 計算器 615<sub>1</sub>乃至 615<sub>5</sub>は、復号途中結果 D614 と D610 と、受信データ D617 とを用いて、制御信号 D622 に基づいて、式 (1)、式 (10)、式 (11) にしたがって、それぞれ演算を行い、その演算の結果得られる検査行列の各列の 1 に対応する復号途中結果 D615( $v_i'$ )を復号途中結果格納用メモリ 610 に供給する。

15 ステップ S71 の処理後は、ステップ S72 に進み、復号途中結果格納用メモリ 610 は、ステップ S71 で計算部 615 から供給された復号途中結果 D615 を、同一アドレスに格納するとともに、既に格納している復号途中結果 D615 (D610) を読み出して、サイクリックシフト回路 611 と計算部 615 に供給する。

20 ステップ S72 の処理後は、ステップ S73 に進み、制御部 617 は、計算部 615 により、検査行列の各列の 1 に対応する全ての復号途中結果 D615 が演算されたかどうかを判定し、全ての復号途中結果 D615 が演算されていないと判定した場合、ステップ S70 に戻り、上述した処理を繰り返す。

一方、ステップ S73 において、制御部 617 は、計算部 615 により、全ての復号途中結果 D615 が演算されたと判定した場合、ステップ S74 に進み、サイクリックシフト回路 611 は、復号途中結果格納用メモリ 610 から供給される復号途中結果 D610( $v_i'$ )をサイクリックシフトする。

具体的には、サイクリックシフト回路 6 1 1 には、復号途中結果格納用メモリ 6 1 0 から 5 つの復号途中結果 D610 が供給されるとともに、制御部 6 1 7 から、その復号途中結果 D610 に対応する検査行列の 1 が検査行列において元となる単位行列などを幾つサイクリックシフトしたものであるかの情報 (Matrix データ) を表す制御信号 D619 が供給される。サイクリックシフト回路 6 1 1 は、制御信号 D619 を元に、5 つの復号途中結果 D610 をサイクリックシフトし (並べ替え)、その結果を復号途中結果 D611 として、計算部 6 1 2 に供給する。

5      ステップ S 7 4 の処理後は、ステップ S 7 5 に進み、計算部 6 1 2 は、第 1 の演算を行い、その演算結果である復号途中結果 D612 をサイクリックシフト回路 6 1 4 に供給する。

具体的には、計算部 6 1 2 には、ステップ S 7 4 でサイクリックシフト回路 6 1 1 から 5 つの復号途中結果  $D611(v_i')$  が供給され、その復号途中結果 D611 が、計算部 6 1 2 の計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> のそれぞれに 1 つずつ供給される。さらに、計算部 6 1 2 には、制御部 6 1 7 から制御信号 D621 が供給され、その制御信号 D621 が計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> に供給される。

15      計算器 6 1 2<sub>1</sub> 乃至 6 1 2<sub>5</sub> は、それぞれ復号途中結果 D611 を用いて、制御信号 D619 に基づいて、式 (9) にしたがって、それぞれ演算を行い、その演算の結果得られる検査行列の行に対応する復号途中結果 D612 (w) を復号途中結果格納用メモリ 6 1 3 に供給する。

20      ステップ S 7 5 の処理後は、ステップ S 7 6 に進み、復号途中結果格納用メモリ 6 1 3 は、ステップ S 7 5 で計算部 6 1 2 から供給された復号途中結果 D612 を、同一アドレスに格納し、ステップ S 7 7 に進む。

ステップ S 7 7 において、制御部 6 1 7 は、計算部 6 1 2 により、検査行列の全ての行に対応する復号途中結果 D612 が演算されたかどうかを判定し、全ての復号途中結果が演算されていないと判定した場合、ステップ S 7 4 に戻り、上述した処理を繰り返す。

一方、ステップ S 7 7 において、制御部 6 1 7 は、計算部 6 1 2 により、全て

の行に対応する復号途中結果 D612 が演算されたと判定した場合、処理を終了する。

なお、復号装置 600 は、復号回数だけ図 33 の復号処理を繰り返し行ない、計算部 621 により上述した式 (5) にしたが演算の結果得られる値 D661 が、

5 最終的な復号結果として出力される。

上述した説明では、復号途中結果格納用メモリ 610 は、2つのシングルポート RAM から構成したが、1つの RAM に対して読み出しと書き込みが同時に起こらないようにすれば、3つ以上の RAM から構成してもよい。例えば、RAM の物理的なビットが足りない場合には、複数の RAM を用いて同じ制御信号を与えるこ

10 とで、論理的に 1つの RAM とみなすことができる。

また、枝データ（枝に対応するメッセージ）が欠けている箇所に関しては、メモリ格納時（復号途中結果格納用メモリ 610 と 613 へのデータ格納時）には、何のメッセージも格納せず、また、演算時（計算部 612 での第 1 の演算時と計算部 615 での第 2 の演算時）にも何の演算も行わない。

15 また、図 16A と図 16B のサイクリックシフト回路 314 および 320、図 18 のサイクリックシフト回路 411 および 414、図 26 のサイクリックシフト回路 611 および 614 には、バレルシフタを用いると回路規模を小さくしながら所望の操作を実現できる。

上述の場合には、説明を簡単にするために、P が 5 の場合、即ち、検査行列を  
20 構成する構成行列の行数および列数が 5 の場合を例に挙げたが、構成行列の行数および列数 P は必ずしも 5 である必要はなく、検査行列によって異なる値を取ることもあり得る。例えば、P は 360 や 392 であってもよい。

また、本実施の形態では、符号長 90、符号化率  $2/3$  の LDPC 符号を用いたが、LDPC 符号の符号長や符号化率は、幾つであっても構わない。例えば、構成行列  
25 の行数および列数 P が 5 の場合、枝総数が 5 以下であれば、どんな符号長、符号化率の LDPC 符号でも、制御信号を代えるだけで、図 16A 乃至図 16C の復号装置 300、図 18 の復号装置 400、図 26 の復号装置 600 を用いて復号

可能である。

さらに、構成行列の行数および列数  $P$  が所定の値で、枝の総数がある値以下、という条件を満たすある LDPC 符号の復号装置は、その条件を満たす、任意の符号長で、任意の符号化率の LDPC 符号を復号することができる。

- 5 検査行列が、構成行列の行数および列数  $P$  の倍数でない場合は、検査行列の端数の外側にすべて 0 (all 0) の成分を付けて  $P$  の倍数とみなして適用できることがある。

次に、上述した一連の処理は、ハードウェアにより行うこともできるし、ソフトウェアにより行うこともできる。一連の処理をソフトウェアによって行う場合  
10 には、そのソフトウェアを構成するプログラムが、汎用のコンピュータ等にインストールされる。

そこで、図 3 4 は、上述した一連の処理を実行するプログラムがインストールされるコンピュータの一実施の形態の構成例を示している。

- 15 プログラムは、コンピュータに内蔵されている記録媒体としてのハードディスク 9 0 5 や ROM 9 0 3 に予め記録しておくことができる。

あるいはまた、プログラムは、フレキシブルディスク、CD-ROM (Compact Disc Read Only Memory)、MO (Magneto Optical) ディスク、DVD (Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体  
9 1 1 に、一時的あるいは永続的に格納（記録）しておくことができる。このよ  
20 うなリムーバブル記録媒体 9 1 1 は、いわゆるパッケージソフトウェアとして提供することができる。

なお、プログラムは、上述したようなリムーバブル記録媒体 9 1 1 からコンピュータにインストールする他、ダウンロードサイトから、デジタル衛星放送用の人工衛星を介して、コンピュータに無線で転送したり、LAN (Local Area  
25 Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを、通信部 9 0 8 で受信し、内蔵するハードディスク 9 0 5 にインストールすること

ができる。

コンピュータは、CPU(Central Processing Unit) 902を内蔵している。CPU 902には、バス901を介して、入出力インタフェース910が接続されており、CPU 902は、入出力インタフェース910を介して、ユーザによって、

5 キーボードや、マウス、マイク等で構成される入力部907が操作等されることにより指令が入力されると、それにしたがって、ROM(Read Only Memory) 903に格納されているプログラムを実行する。あるいは、また、CPU 902は、ハードディスク905に格納されているプログラム、衛星若しくはネットワークから転送され、通信部908で受信されてハードディスク905にインストールされ

10 たプログラム、またはドライブ909に装着されたリムーバブル記録媒体911から読み出されてハードディスク905にインストールされたプログラムを、RAM(Random Access Memory) 904にロードして実行する。これにより、CPU 902は、上述したフローチャートにしたがった処理、あるいは上述したブロック図の構成により行われる処理を行う。そして、CPU 902は、その処理結果を、

15 必要に応じて、例えば、入出力インタフェース910を介して、LCD(Liquid Crystal Display)やスピーカ等で構成される出力部906から出力、あるいは、通信部908から送信、さらには、ハードディスク905に記録等させる。

ここで、本明細書において、コンピュータに各種の処理を行わせるためのプログラムを記述する処理ステップは、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいは個別に実行される処理

20 (例えば、並列処理あるいはオブジェクトによる処理)も含むものである。

また、プログラムは、1のコンピュータにより処理されるものであっても良いし、複数のコンピュータによって分散処理されるものであっても良い。さらに、プログラムは、遠方のコンピュータに転送されて実行されるものであっても良い。

25 以上のように、 $P \times P$ の単位行列、そのコンポーネントの1のうち1個以上が0になった準単位行列、単位行列もしくは準単位行列をサイクリックシフトしたシフト行列、単位行列、準単位行列、もしくはシフト行列の複数の和である和行

列、 $P \times P$  の  $Q$  行列の組合せで表わすことができる検査行列を持つ LDPC 符号の復号を、チェックノードとバリアブルノードの演算を  $P$  個同時に行うアーキテクチャ (architecture) を採用することにより、ノード演算を、 $P$  個同時に行うことで動作周波数を実現可能な範囲に抑えることができ、多数の繰り返し復号を行うことを可能にしつつ、メモリ (FIFO や RAM) への書き込みと読み出し時に、異なるアドレスへの同時アクセスが起きることを防止することができる。

さらに、図 16 A 乃至図 16 C の復号装置 300 を繰り返し用いて、図 15 の検査行列で表わされる LDPC 符号を復号する場合には、269 個の枝をチェックノード、バリアブルノード毎に 5 個ずつ演算することが可能であることから、10 回の復号に、 $269/5 \times 2 = 108$  クロック動作すればよいことになる。50 回の復号には、90 個の符号情報を受信する間に、 $108 \times 50 = 5400$  クロック動作すればよいことになり、受信周波数の約 60 倍の動作周波数でよいことになる。従って、図 16 A 乃至図 16 C の復号装置 300 によれば、各ノード演算を一つずつ行う図 9 の復号装置に比べて、 $1/5$  の動作周波数で済むことになる。また、回路規模の面から見ても、メモリの大きさは同じであるため、論理回路が多少大きくなっても全体への影響は小さいと言える。

さらに、図 18 の復号装置 400 と図 26 の復号装置 600 は、図 16 A 乃至図 16 C の復号装置 300 に比べて、メモリの容量が小さくなっている。

例えば、LDPC 符号の検査行列が図 15 の検査行列であり、LDPC 符号の量子化ビット数が 6 ビットである場合、図 16 A 乃至図 16 C の復号装置 300 では、枝データ格納メモリに、全枝数の  $269 \times 6 = 1614$  ビットの容量を有する RAM 2 つ、即ち、2 つの RAM で  $1614 \times 2 = 3228$  ビットの容量が必要であった。これに対して、例えば、復号途中結果  $v$  の量子化ビット数が 9 ビットである場合、図 18 の復号装置 400 では、復号途中結果格納用メモリ 413 に、全枝数の 1614 ビットの容量を有する RAM と、復号途中結果格納用メモリ 410 に、LDPC 符号の符号長 (検査行列の列数) と復号途中結果  $v$  の量子化ビット数との乗算値、即ち  $90 \times 9 = 810$  ビットの容量を有する RAM を備えれば

よく、復号装置の回路規模を小さくすることができる。さらに、図18の復号装置400では、第2の演算を行う計算部415において、FIFOメモリを有する必要がないので、ロジックの回路規模を小さくすることができる。

また、例えば、LDPC符号の検査行列が図15の検査行列で、復号途中結果v  
5 の量子化ビット数が10ビットである場合、図26の復号装置600では、復号途中結果格納用メモリ610に、全枝数の1614ビットの容量を有するRAMと、復号途中結果格納用メモリ613に、検査行列の行数と復号途中結果vとの乗算値、即ち $30 \times 10 = 300$ ビットの容量を有するRAMを備えればよく、復号装置の回路規模を小さくすることができる。さらに、図26の復号装置60  
10 0では、第1の演算を行う計算部612において、FIFOメモリを有する必要がないので、ロジックの回路規模を小さくすることができる。

一般的に、LDPC符号は符号長が数千から数万と大きいので、Pの値も数百の大きさを持つものが使われる。その場合には、更に本発明に係る復号装置を用いる効果は大きくなる。

15 また、本発明に係る復号装置は、サンプラダクトアルゴリズムを忠実に実装するものであるため、メッセージの量子化以外の復号損失が起きることはない。

#### 産業上の利用可能性

以上の観点から、本発明に係る復号装置を用いることで、高性能な復号が可能  
20 になる。

## 請求の範囲

1. LDPC (Low Density Parity Check) 符号の復号装置であって、

$P \times P$  の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または  $P \times P$  の 0 行列を構成行列として、前記 LDPC 符号の検査行列が、複数の前記構成行列の組合せで表される場合において、

10 前記 LDPC 符号の復号のための  $P$  個のチェックノードの演算を同時に行う第 1 の演算手段と、

前記 LDPC 符号の復号のための  $P$  個のバリエブルノードの演算を同時に行う第 2 の演算手段と

を備えることを特徴とする復号装置。

2. 請求の範囲第 1 項に記載の復号装置であって、

15 前記第 1 の演算手段は、チェックノードの演算を行う  $P$  個のチェックノード計算器を有し、

前記第 2 の演算手段は、バリエブルノードの演算を行う  $P$  個のバリエブルノード計算器を有する

ことを特徴とする復号装置。

20 3. 請求の範囲第 1 項に記載の復号装置であって、

前記  $P$  個のチェックノードの演算、または前記  $P$  個のバリエブルノードの演算の結果得られる  $P$  個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段をさらに備える

ことを特徴とする復号装置。

25 4. 請求の範囲第 3 項に記載の復号装置であって、

前記メッセージ記憶手段は、チェックノードの演算時に読み出される枝に対応するメッセージデータを、検査行列の 1 を行方向に詰めるように格納する



ことを特徴とする復号装置。

5. 請求の範囲第3項に記載の復号装置であって、

前記メッセージ記憶手段は、バリアブルノード演算時に読み出される枝に対応するメッセージデータを、検査行列の1を列方向に詰めるように格納する

5 ことを特徴とする復号装置。

6. 請求の範囲第3項に記載の復号装置であって、

前記メッセージ記憶手段は、前記検査行列を表す構成行列のうちの、重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位  
10 行列、またはシフト行列に属するP個の枝に対応するメッセージを、同一のアドレスに格納する

ことを特徴とする復号装置。

7. 請求の範囲第3項に記載の復号装置であって、

前記メッセージ記憶手段は、行数/P個のFIFOと、列数/P個のFIFOとで構  
15 成され、

前記行数/P個のFIFOと列数/P個のFIFOは、それぞれ、前記検査行列の行と列の重みに対応するワード数を有する

ことを特徴とする復号装置。

8. 請求の範囲第3項に記載の復号装置であって、

20 前記メッセージ記憶手段は、RAM(Random Access Memory)で構成され

前記RAMは、前記メッセージデータを、読み出される順番に詰めて格納し、格納位置順に読み出す

ことを特徴とする復号装置。

9. 請求の範囲第1項に記載の復号装置であって、

25 LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み出す受信情報記憶手段をさらに備える

ことを特徴とする復号装置。

10. 請求の範囲第9項に記載の復号装置であって、

前記受信情報記憶手段は、前記受信情報を、前記バリエブルノードの演算に必要となる順番に読み出すことができるように格納する

ことを特徴とする復号装置。

5 11. 請求の範囲第1項に記載の復号装置であって、

前記P個のチェックノードの演算、または前記P個のバリエブルノードの演算の結果得られるメッセージを並べ替える並べ替え手段をさらに備える

ことを特徴とする復号装置。

12. 請求の範囲第11項に記載の復号装置であって、

10 前記並べ替え手段は、バレルシフタで構成される

ことを特徴とする復号装置。

13. 請求の範囲第1項に記載の復号装置であって、

前記第1の演算手段と前記第2の演算手段は、P個の枝に対応するメッセージを求める

15 ことを特徴とする復号装置。

14. 請求の範囲第1項に記載の復号装置であって、

前記第1の演算手段は、前記P個のチェックノードの演算と前記P個のバリエブルノードの演算の一部とを行い、

前記第2の演算手段は、前記P個のバリエブルノードの演算の他の一部を行

20 う

ことを特徴とする復号装置。

15. 請求の範囲第14項に記載の復号装置であって、

前記第1の演算手段は、前記P個のチェックノードの演算と前記P個のバリエブルノードの演算の一部を行うP個の計算器を有し、

25 前記第2の演算手段は、前記P個のバリエブルノードの演算の他の一部を行うP個の計算器を有する

ことを特徴とする復号装置。

16. 請求の範囲第14項に記載の復号装置であって、

前記第1の演算手段が前記P個のチェックノードの演算と前記P個のバリエブルノードの演算の一部を行うことにより得られるP個の枝に対応する第1の復号途中結果を同時に読み書きする第1の復号途中結果記憶手段をさらに備える

5 ことを特徴とする復号装置。

17. 請求の範囲第16項に記載の復号装置であって、

前記第1の復号途中記憶手段は、前記P個のバリエブルノードの演算の他の一部を行う時に読み出される枝に対応する前記第1の復号途中結果を、検査行列の1を行方向に詰めるように格納する

10 ことを特徴とする復号装置。

18. 請求の範囲第16項に記載の復号装置であって、

前記第1の復号途中結果記憶手段は、2個のシングルポートRAM(Random Access Memory)である

ことを特徴とする復号装置。

15 19. 請求の範囲第18項に記載の復号装置であって、

前記2個のシングルポートRAMは、前記第1の復号途中結果を前記検査行列のP行の枝に対応する前記第1の復号途中結果ずつ交互に格納する

ことを特徴とする復号装置。

20. 請求の範囲第18項に記載の復号装置であって、

20 前記2個のシングルポートRAM(Random Access Memory)は、それぞれ同一のアドレスに格納している前記第1の復号途中結果を読み出す

ことを特徴とする復号装置。

21. 請求の範囲第16項記載の復号装置であって、

前記第1の復号途中結果記憶手段は、前記検査行列を表す構成行列のうちの、

25 重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列に属するP個の枝に対応する前記第1の復号途

中結果を、同一のアドレスに格納する

ことを特徴とする復号装置。

22. 請求の範囲第14項に記載の復号装置であって、

前記第2の演算手段が前記P個のバリエブルノードの演算の他の一部を行う

5 ことにより得られるP個の枝に対応する前記第2の復号途中結果を同時に読み書きする第2の復号途中結果記憶手段をさらに備える

ことを特徴とする復号装置。

23. 請求の範囲第14項に記載の復号装置であって、

LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み

10 出す受信情報記憶手段をさらに備える

ことを特徴とする復号装置。

24. 請求の範囲第23項に記載の復号装置であって、

前記受信情報記憶手段は、前記受信情報を、前記P個のバリエブルノードの演算の他の一部の演算に必要となる順番に読み出すことができるように格納する

15 ことを特徴とする復号装置。

25. 請求の範囲第14項に記載の復号装置であって、

前記第1の演算手段が前記P個のチェックノードの演算と前記P個のバリエブルノードの演算の一部を行うことにより得られる第1の復号途中結果、または前記第2の演算手段が前記P個のバリエブルノードの演算の他の一部を行うこ

20 とにより得られる第2の復号途中結果を並べ替える並べ替え手段をさらに備えることを特徴とする復号装置。

26. 請求の範囲第25項に記載の復号装置であって、

前記並べ替え手段は、バレルシフタで構成される

ことを特徴とする復号装置。

25 27. 請求の範囲第1項に記載の復号装置であって、

前記第1の演算手段は、前記P個のチェックノードの演算の一部を行い、

前記第2の演算手段は、前記P個のチェックノードの演算の他の一部と、前

記 P 個のバリエブルノードの演算とを行う

ことを特徴とする復号装置。

28. 請求の範囲第 27 項に記載の復号装置であって、

前記第 1 の演算手段は、前記 P 個のチェックノードの演算の一部を行う P 個  
5 の計算器を有し、

前記第 2 の演算手段は、前記 P 個のチェックノードの演算の他の一部と、前  
記 P 個のバリエブルノードの演算を行う P 個の計算器を有する

ことを特徴とする復号装置。

29. 請求の範囲第 27 項に記載の復号装置であって、

10 前記第 1 の演算手段が前記 P 個のチェックノードの演算の一部を行うことに  
より得られる P 個の枝に対応する第 1 の復号途中結果を同時に読み書きする第  
1 の復号途中結果記憶手段をさらに備える

ことを特徴とする復号装置。

30. 請求の範囲第 27 項に記載の復号装置であって、

15 前記第 2 の演算手段が前記 P 個のチェックノードの演算の他の一部と、前記 P  
個のバリエブルノードの演算を行うことにより得られる P 個の枝に対応する第  
2 の復号途中結果を同時に読み書きする第 2 の復号途中結果記憶手段をさらに備  
える

ことを特徴とする復号装置。

20 31. 請求の範囲第 30 項に記載の復号装置であって、

前記第 2 の復号途中結果記憶手段は、前記 P 個のチェックノードの演算の他  
の一部と、前記 P 個のバリエブルノードの演算を行う時に読み出される枝に対  
応する前記第 2 の復号途中結果を、検査行列の 1 を列方向に詰めるように格納す  
る

25 ことを特徴とする復号装置。

32. 請求の範囲第 30 項に記載の復号装置であって、

前記第 2 の復号途中結果記憶手段は、2 個のシングルポート RAM(Random

Access Memory)である

ことを特徴とする復号装置。

33. 請求の範囲第32項に記載の復号装置であって、

前記2個のシングルポートRAMは、前記第2の復号途中結果を前記検査行列

5 のP列の枝に対応する前記第2の復号途中結果ずつ交互に格納する

ことを特徴とする復号装置。

34. 請求の範囲第32項に記載の復号装置であって、

前記2個のシングルポートRAM(Random Access Memory)は、それぞれ同一  
のアドレスに格納している前記第2の復号途中結果を読み出す

10 ことを特徴とする復号装置。

35. 請求の範囲第30項に記載の復号装置であって、

前記第2の復号途中結果記憶手段は、前記検査行列を表す構成行列のうちの、  
重みが2以上の構成行列について、その構成行列を、重みが1の単位行列、準単  
位行列、またはシフト行列の和の形で表現したときの、その重みが1の単位行列、  
15 準単位行列、またはシフト行列に属するP個の枝に対応する前記第2の復号途  
中結果を、同一のアドレスに格納する

ことを特徴とする復号装置。

36. 請求の範囲第27項に記載の復号装置であって、

LDPC符号の受信情報を格納するとともに、P個の前記受信情報を同時に読み  
20 出す受信情報記憶手段をさらに備える

ことを特徴とする復号装置。

37. 請求の範囲第36項に記載の復号装置であって、

前記受信情報記憶手段は、前記受信情報を、前記P個のチェックノードの演  
算の他の一部と、前記P個のバリアブルノードの演算に必要なとなる順番に読み  
25 出すことができるように格納する

ことを特徴とする復号装置。

38. 請求の範囲第27項に記載の復号装置であって、

前記第 1 の演算手段が前記 P 個のチェックノードの演算の一部を行うことにより得られる第 1 の復号途中結果、または前記第 2 の演算が前記 P 個のチェックノードの演算の他の一部と、前記 P 個のバリエブルノードの演算を行うことにより得られる第 2 の復号途中結果を並べ替える並べ替え手段をさらに備える

5      ことを特徴とする復号装置。

39.      請求の範囲第 38 項に記載の復号装置であって、

前記並べ替え手段は、バレルシフタで構成される

ことを特徴とする復号装置。

40.      LDPC (Low Density Parity Check) 符号の復号装置の復号方法で

10      あって、

P×P の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または P×P の 0 行列を構成行列として、前記 LDPC 符号の検査行列が、複数の前記構成行列の組合せで表される場合において、

前記 LDPC 符号の復号のための P 個のチェックノードの演算を同時に行う第 1 の演算ステップと、

前記 LDPC 符号の復号のための P 個のバリエブルノードの演算を同時に行う第

20      2 の演算ステップと

を含むことを特徴とする復号方法。

41.      LDPC (Low Density Parity Check) 符号の復号をコンピュータに行わせるプログラムであって、

前記 LDPC 符号の復号のための P 個のチェックノードの演算を同時に行う第 1  
25      の演算ステップと、

前記 LDPC 符号の復号のための P 個のバリエブルノードの演算を同時に行う第 2 の演算ステップと

を含むことを特徴とするプログラム。



1/35

図 1

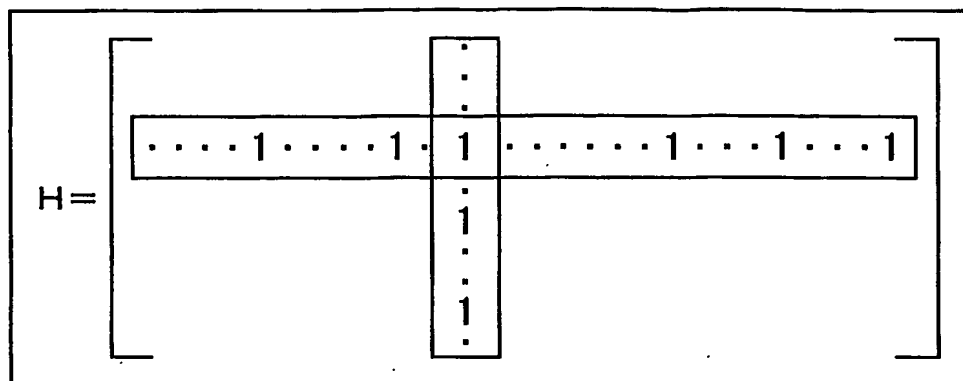
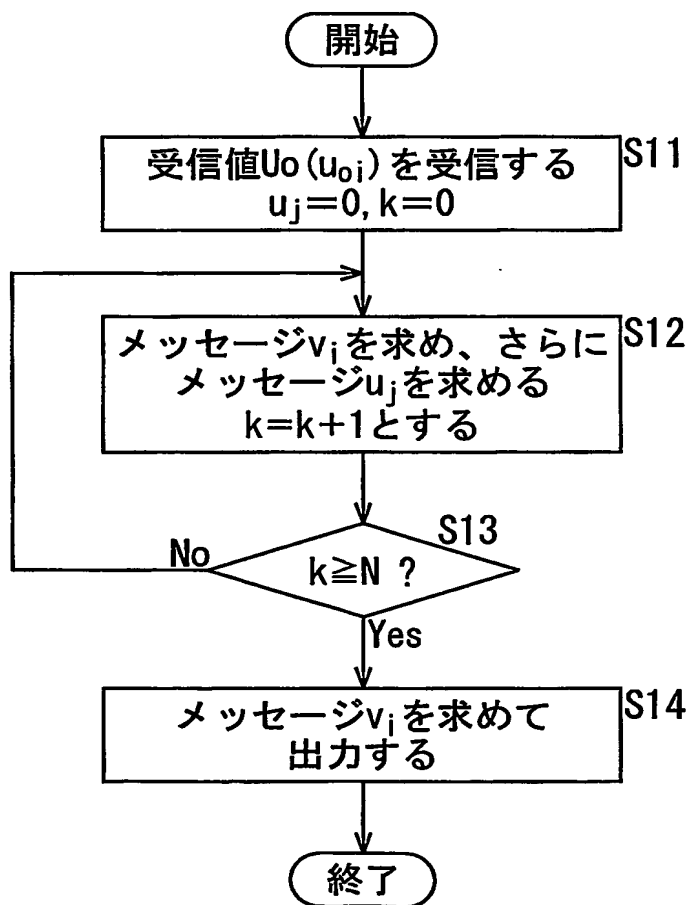


図 2



2/35

図 3

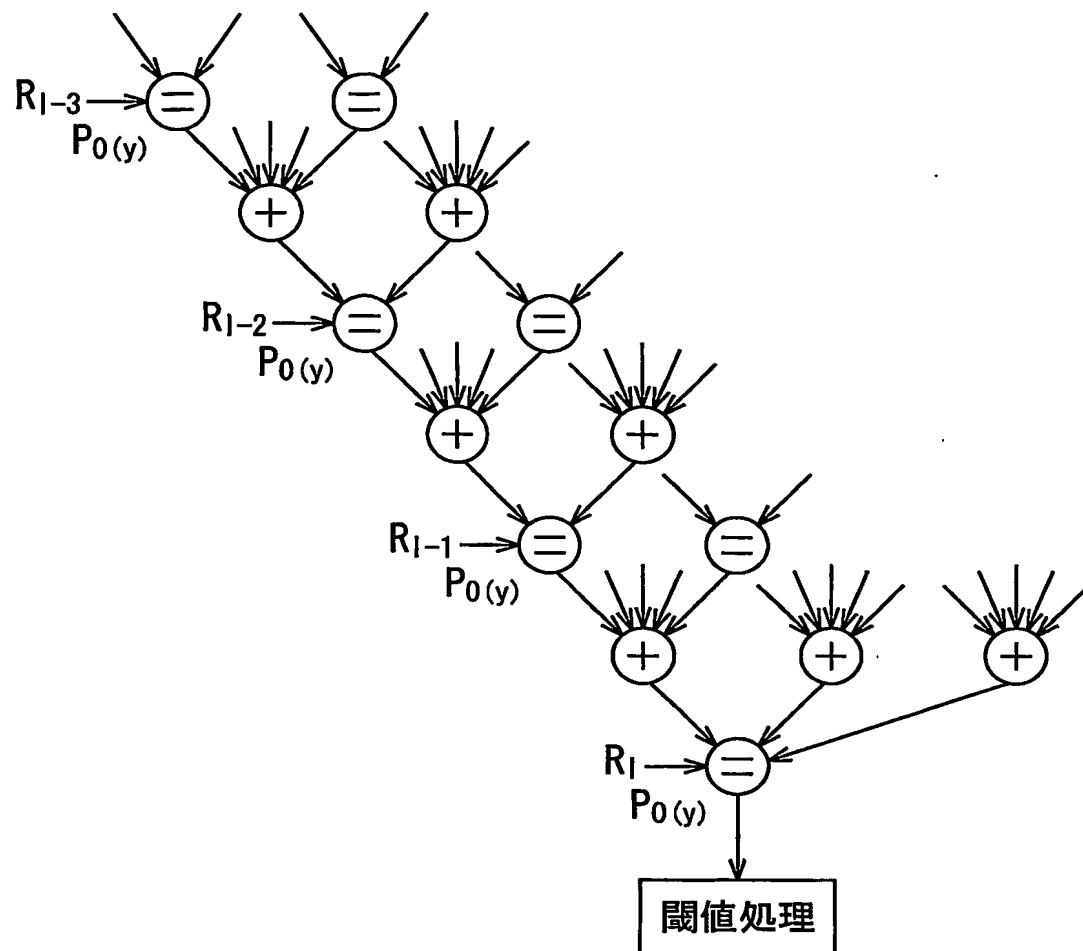
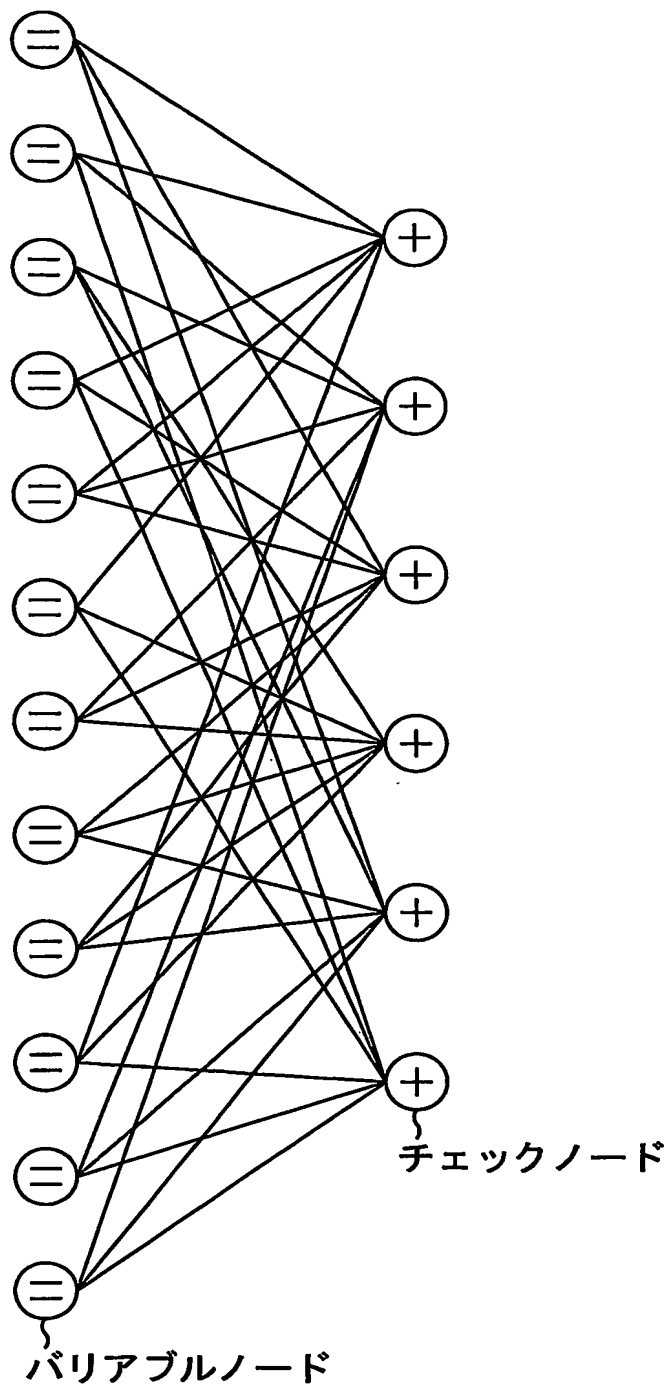


図 4

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

3/35

図 5



4/35

図 6

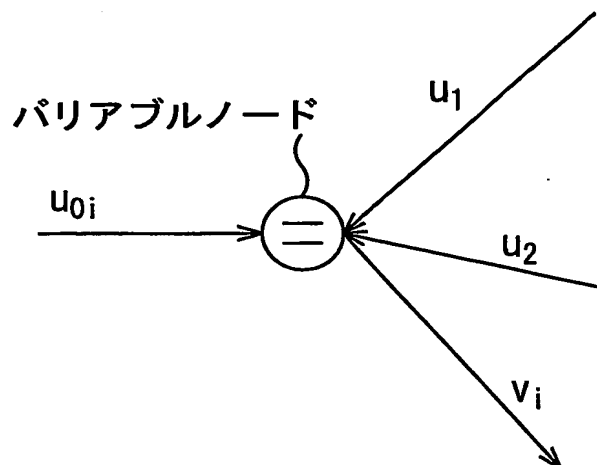
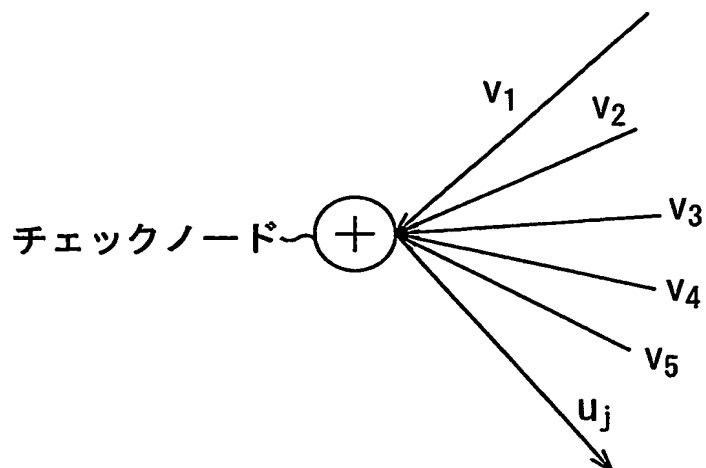


図 7



5/35

図 8

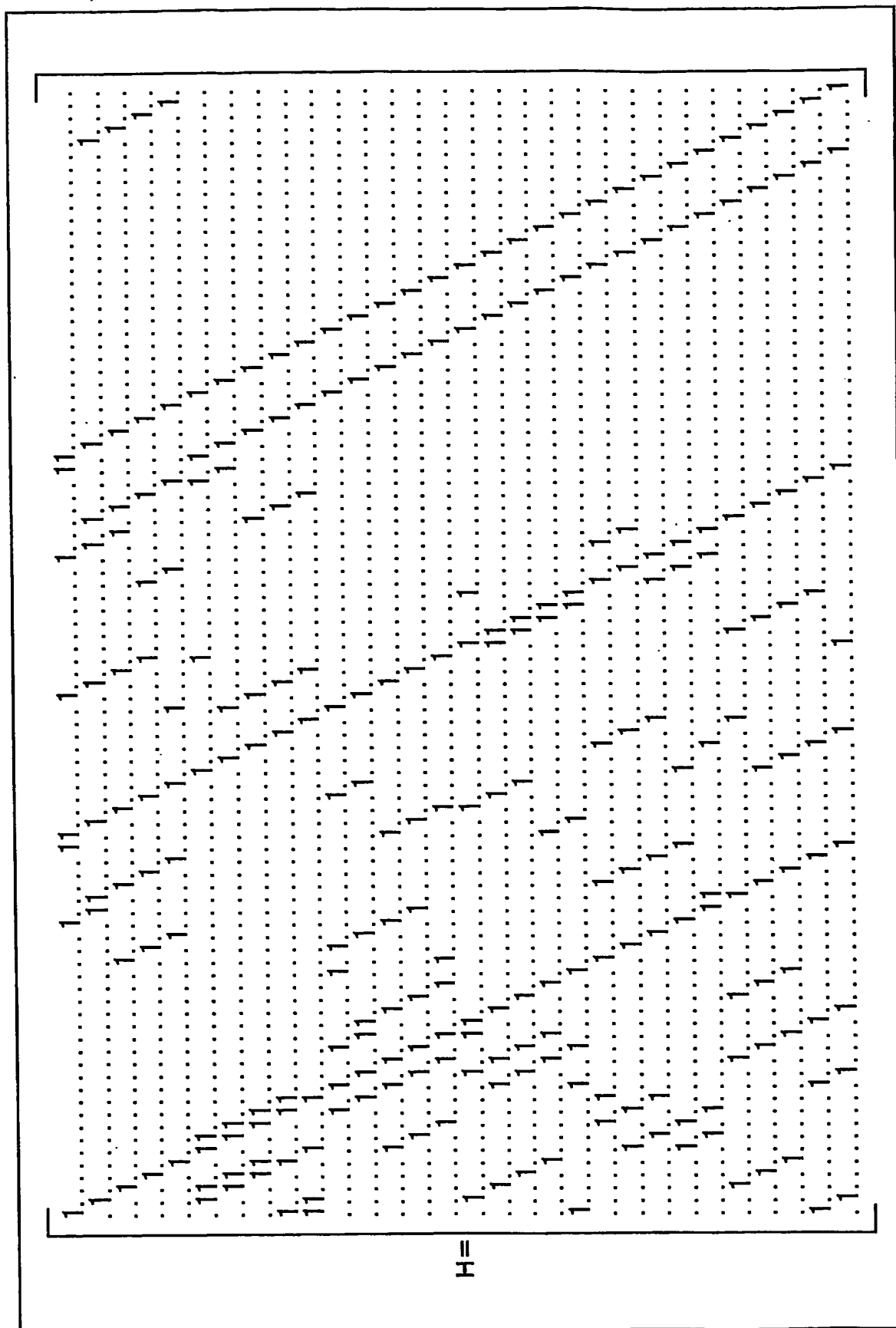


図 9

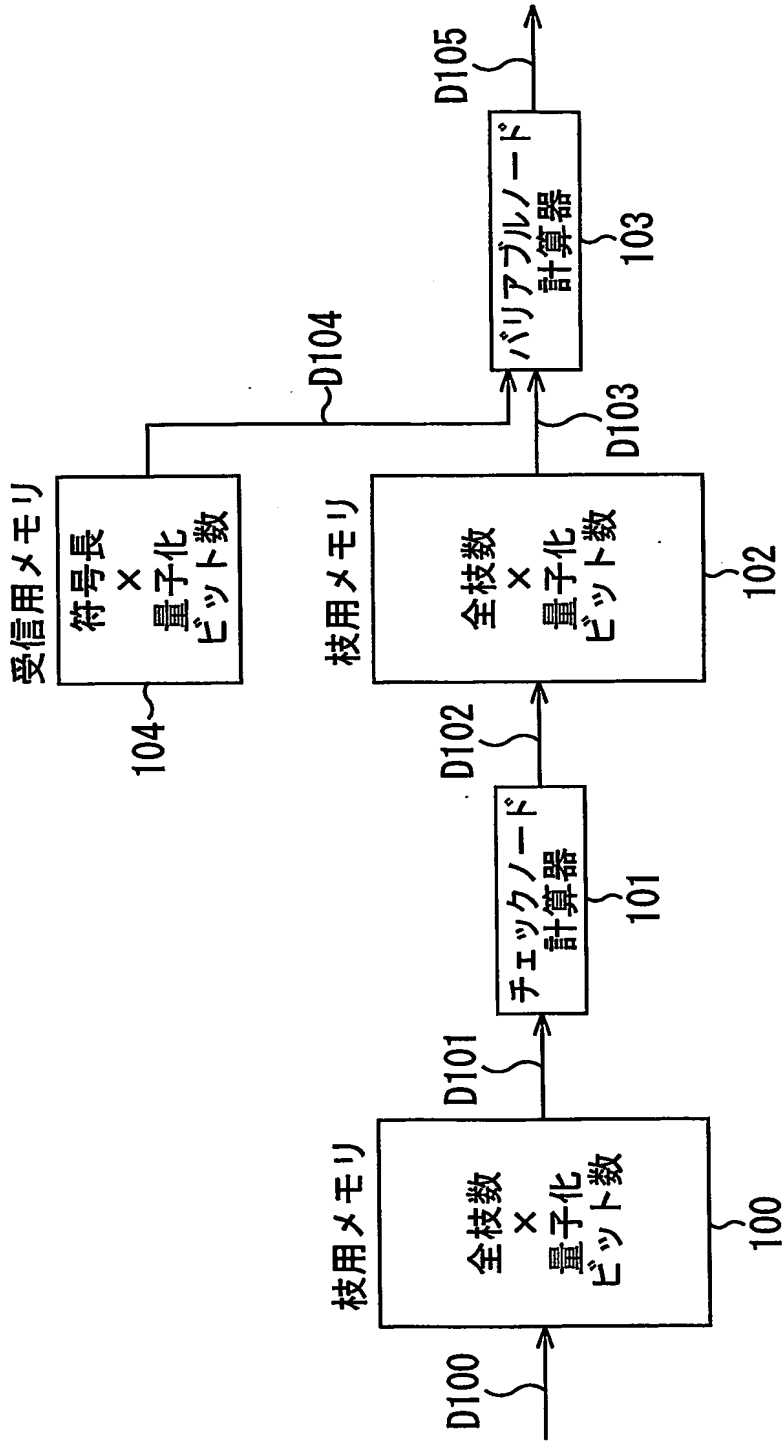


図10

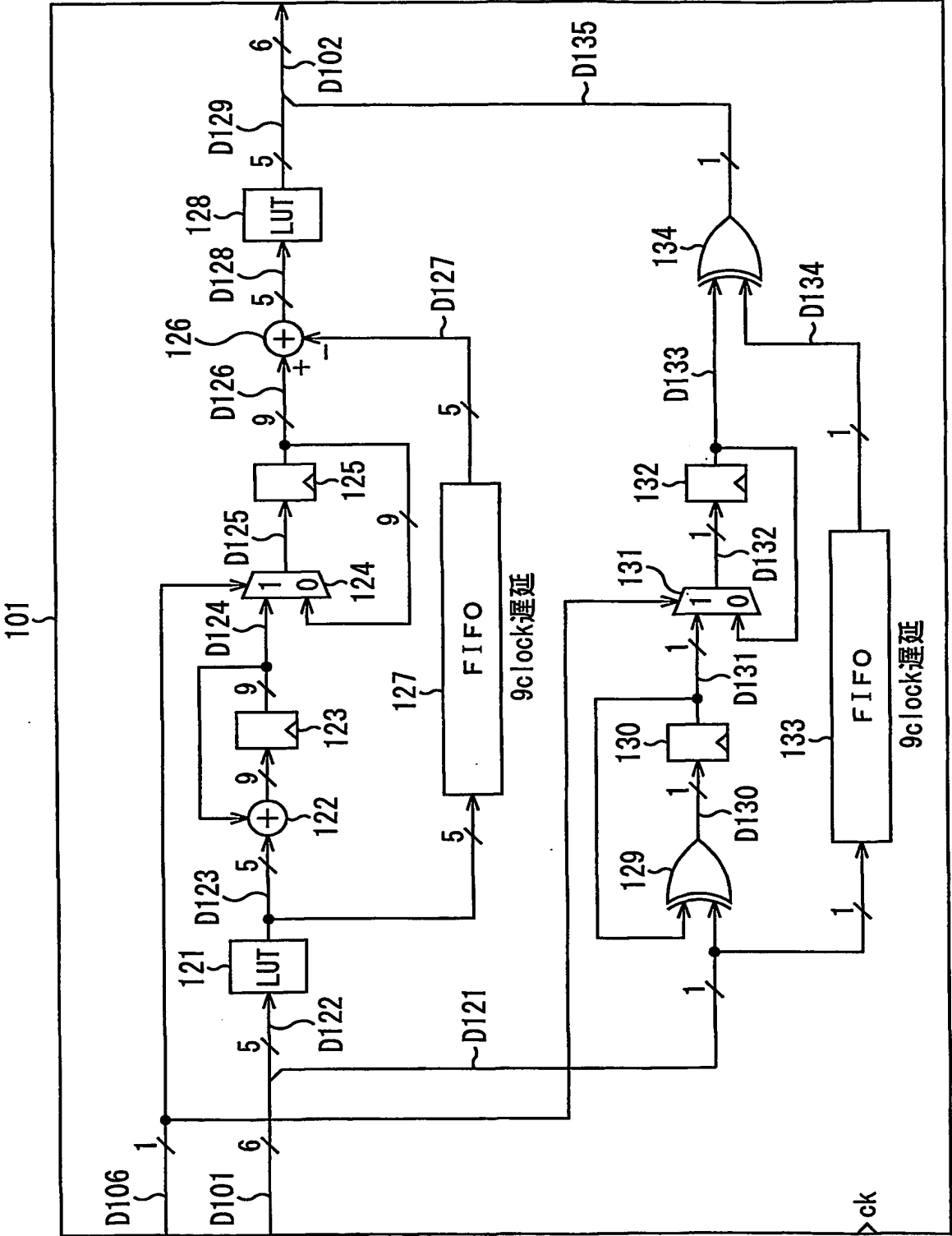


図11

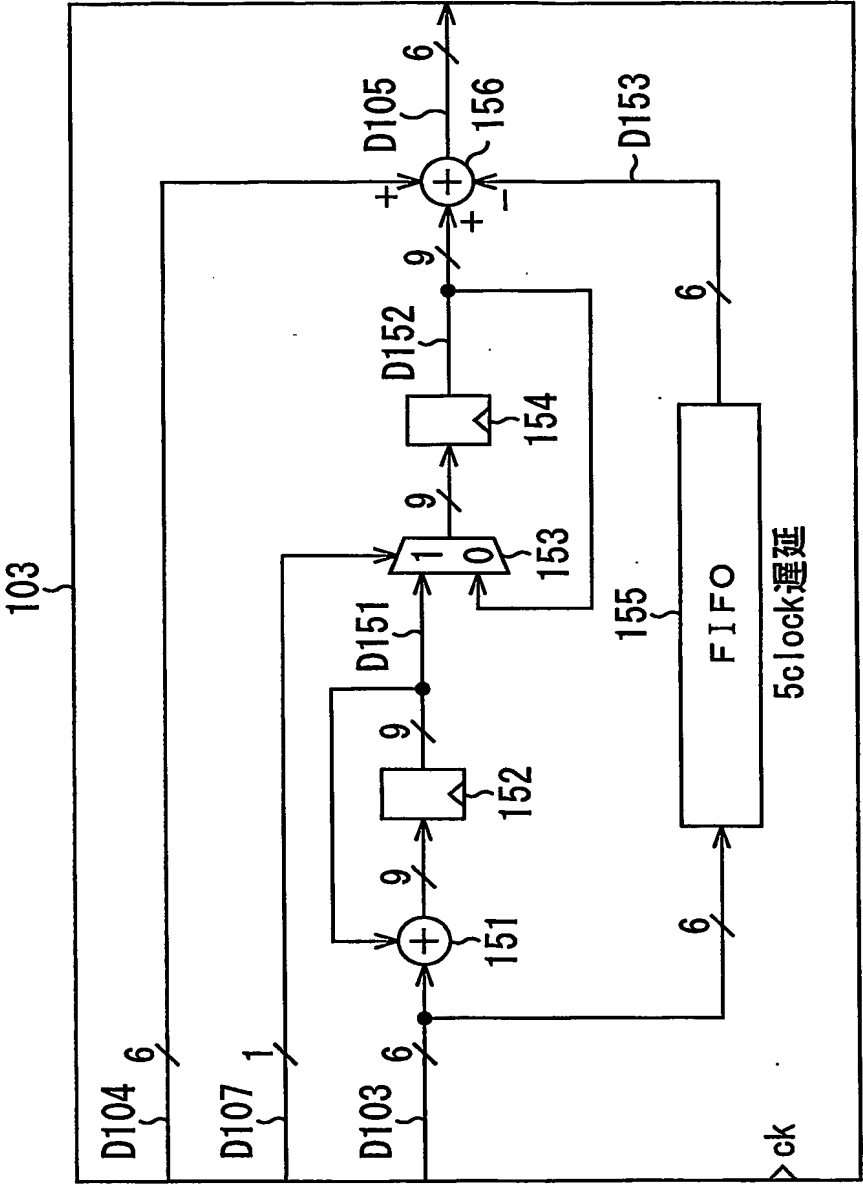




図12A

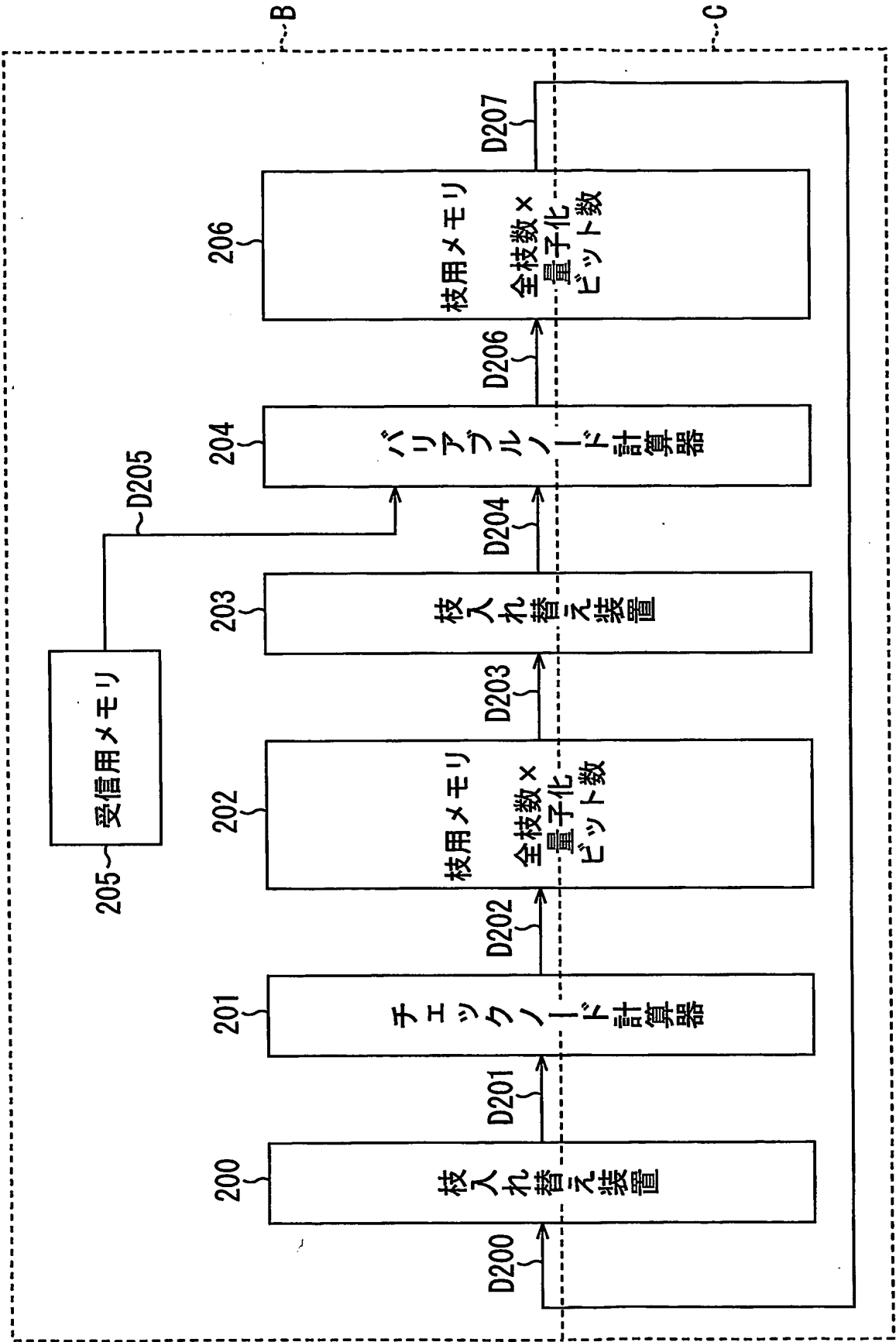


图 12B

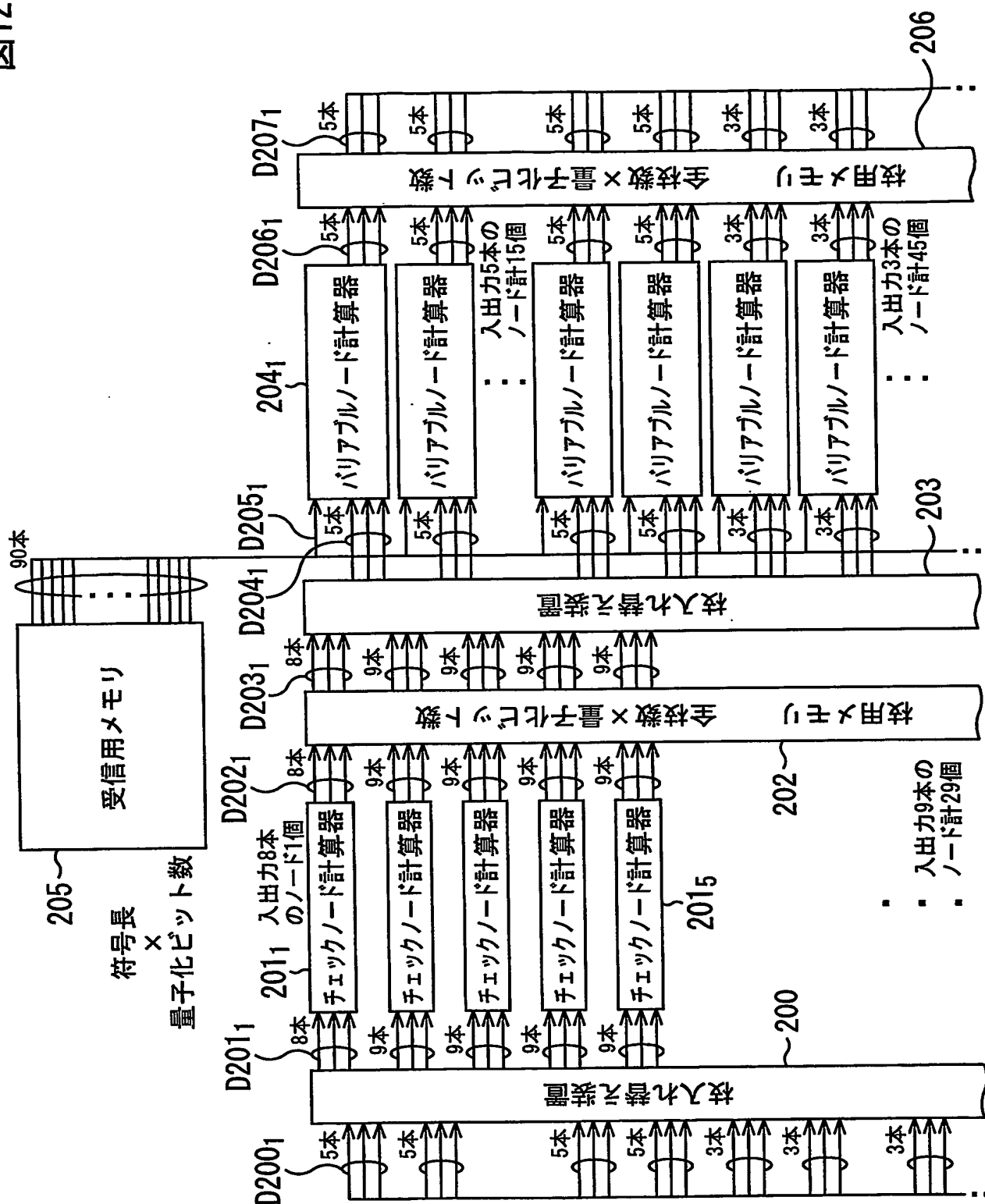


図12C

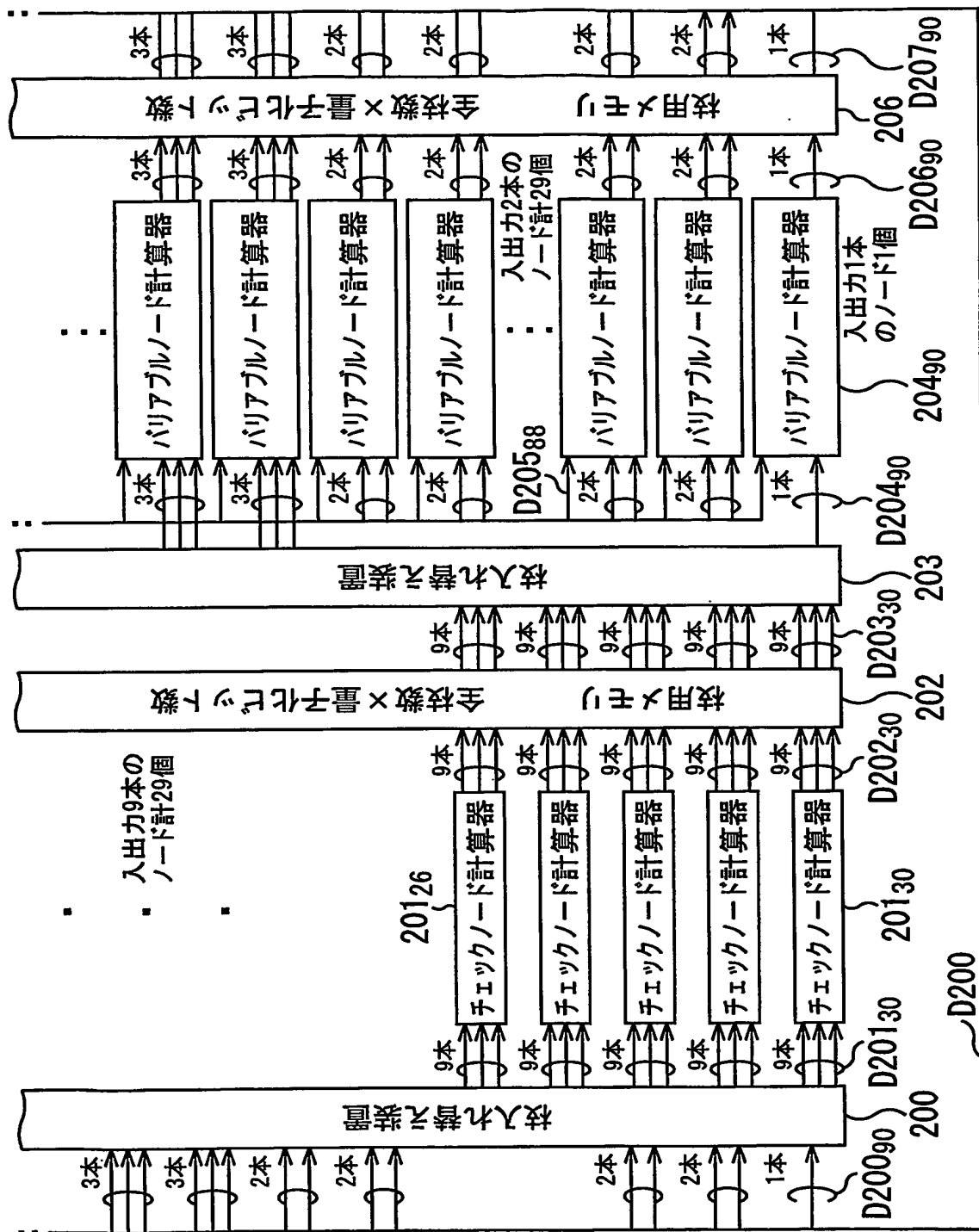


図13

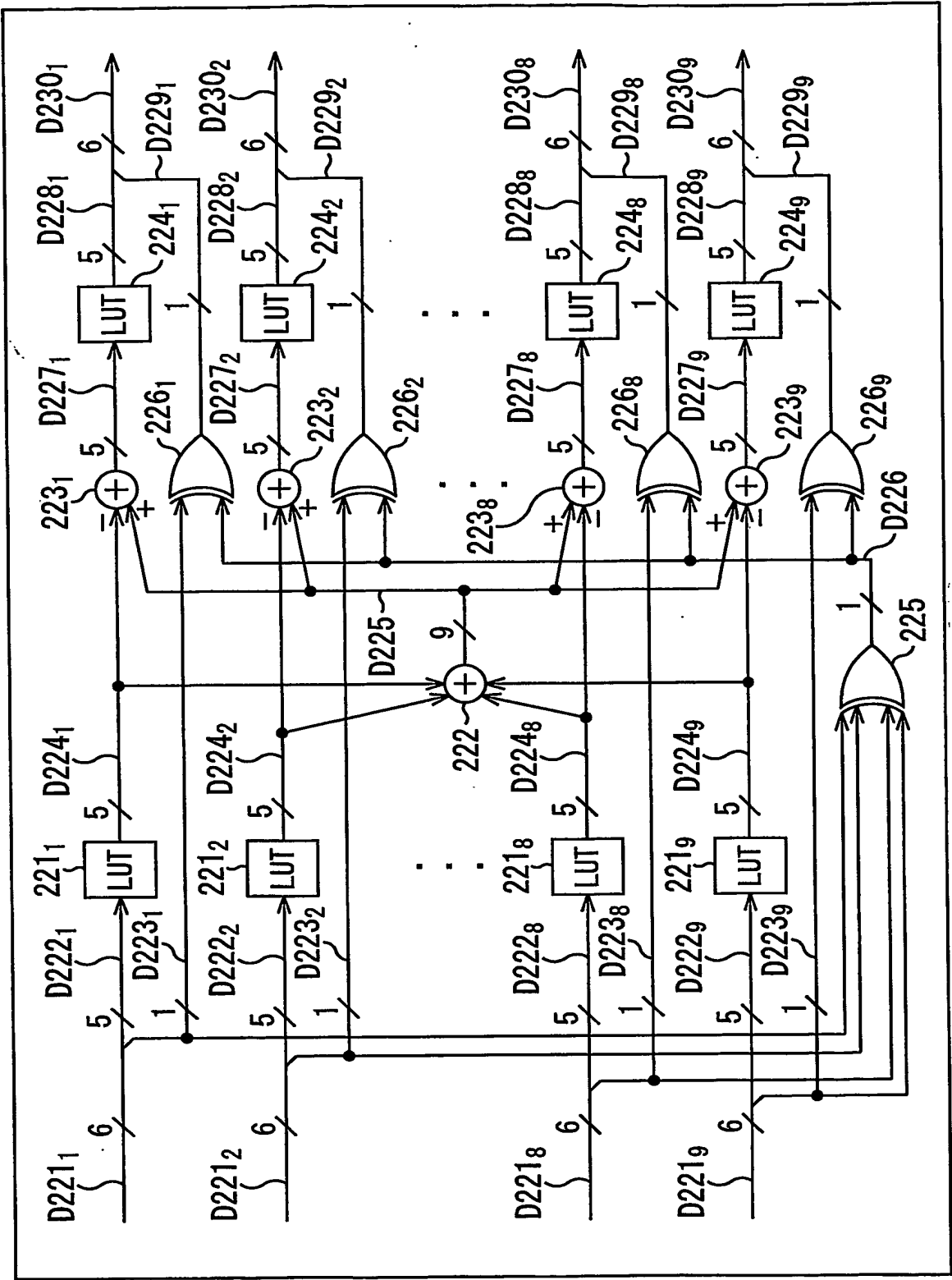
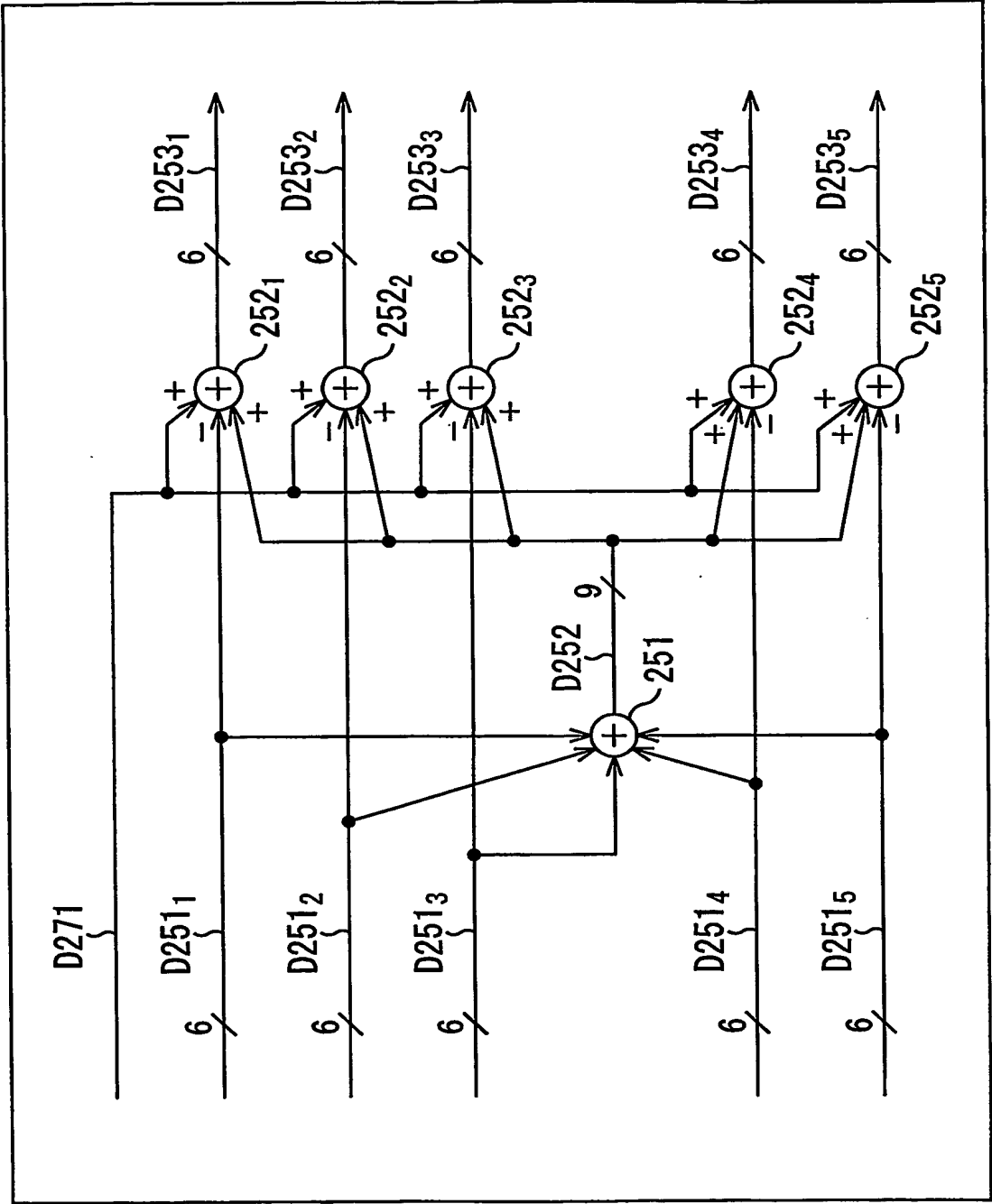
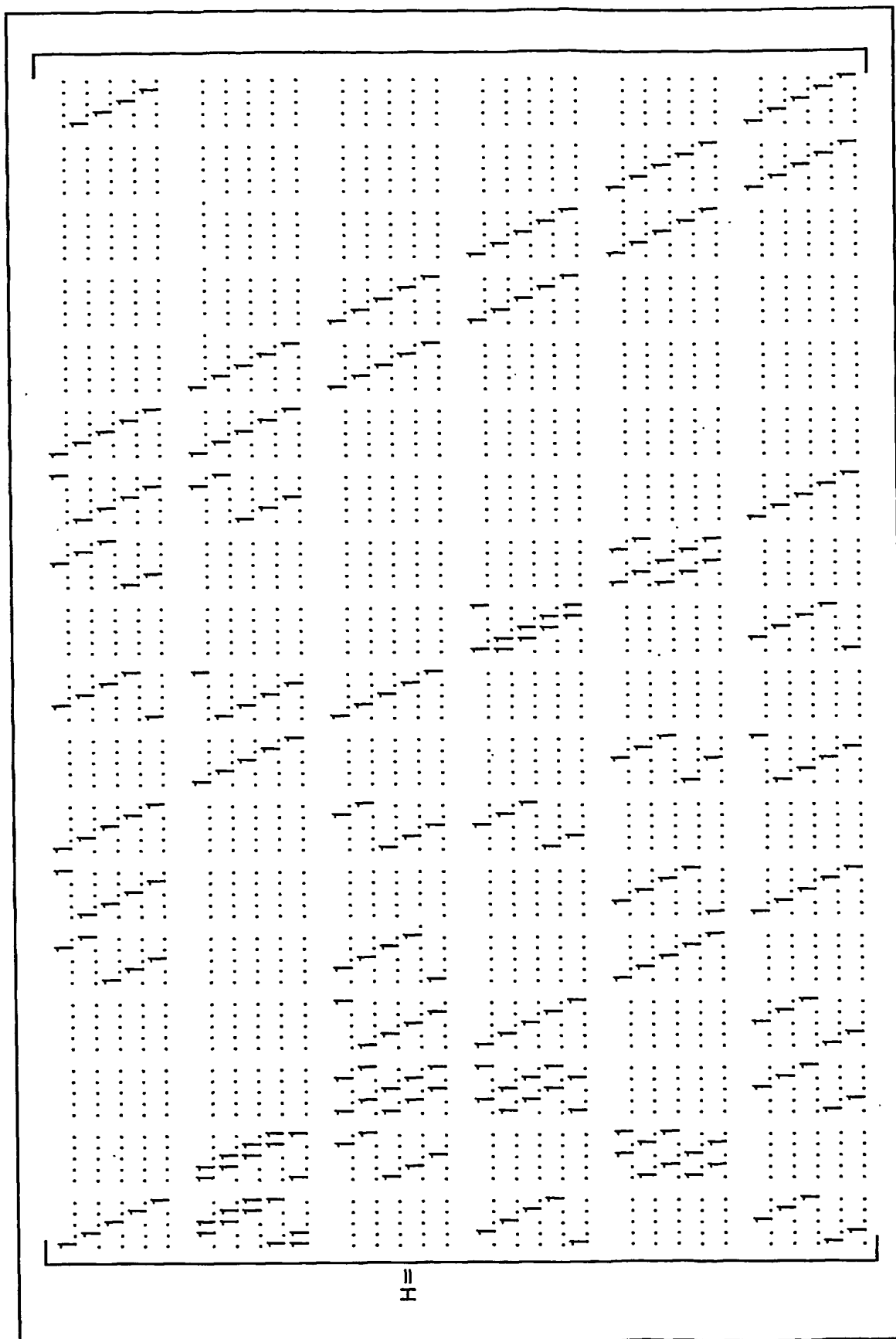


图14



14/35

図15



15/35

図16A

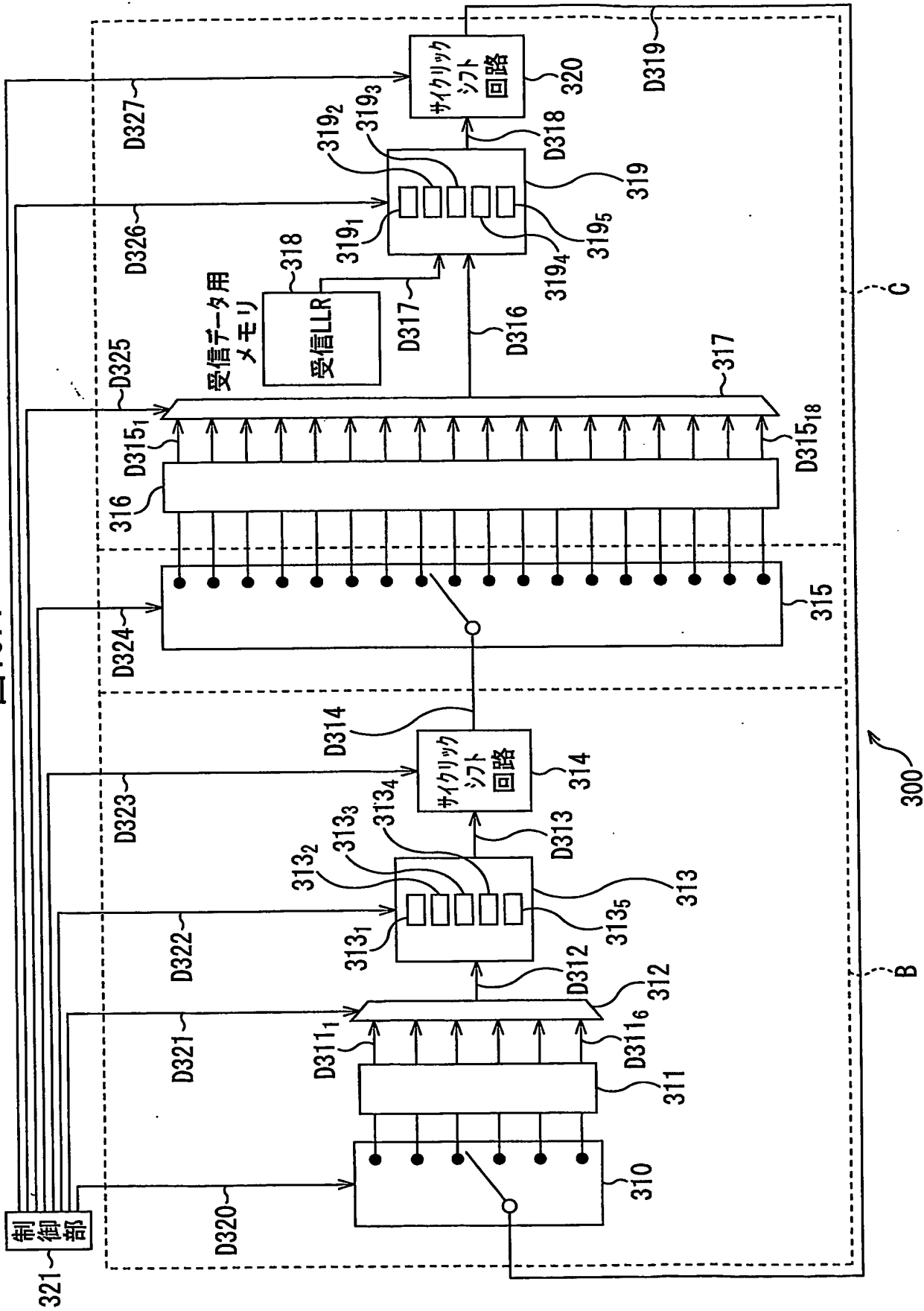
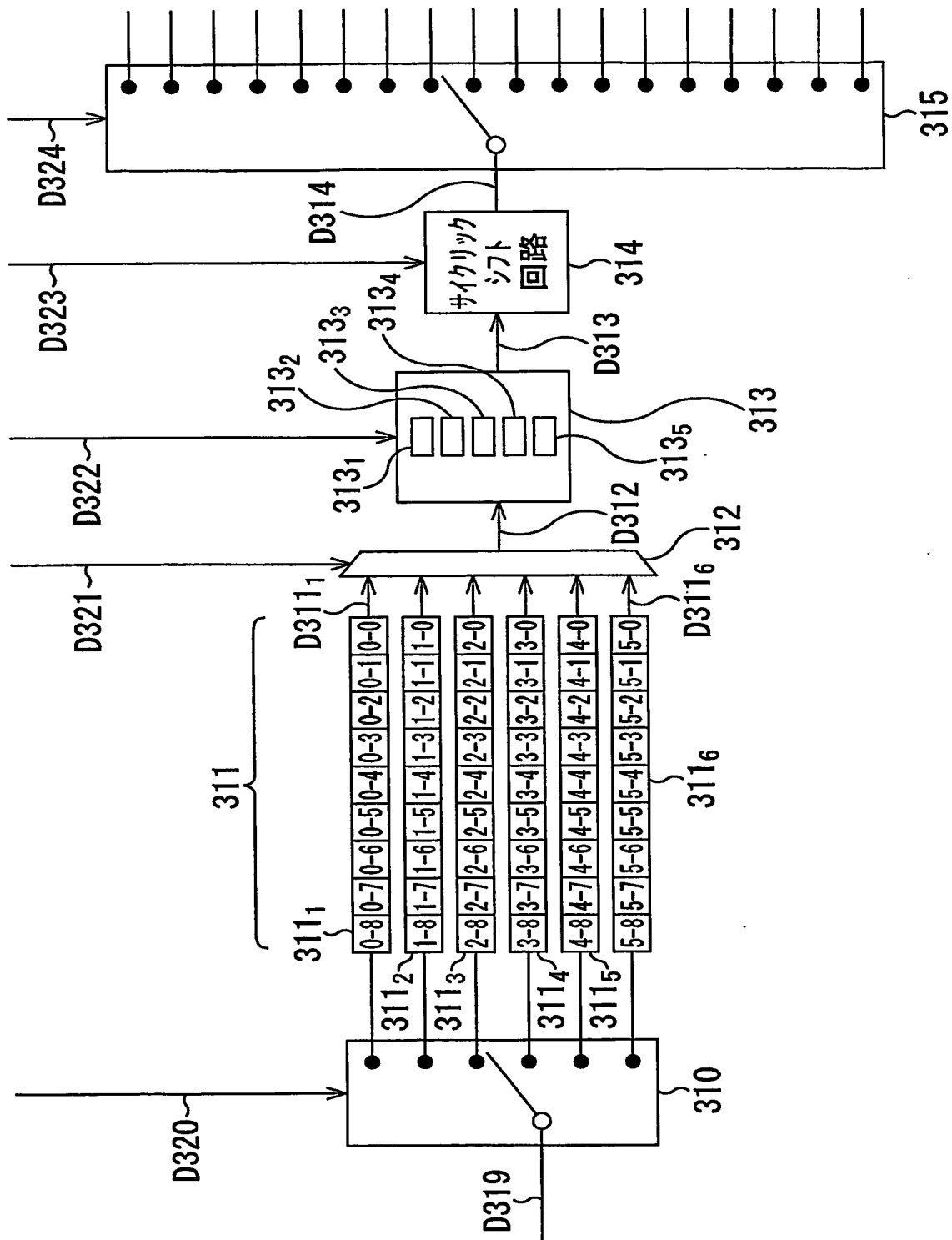


図16B

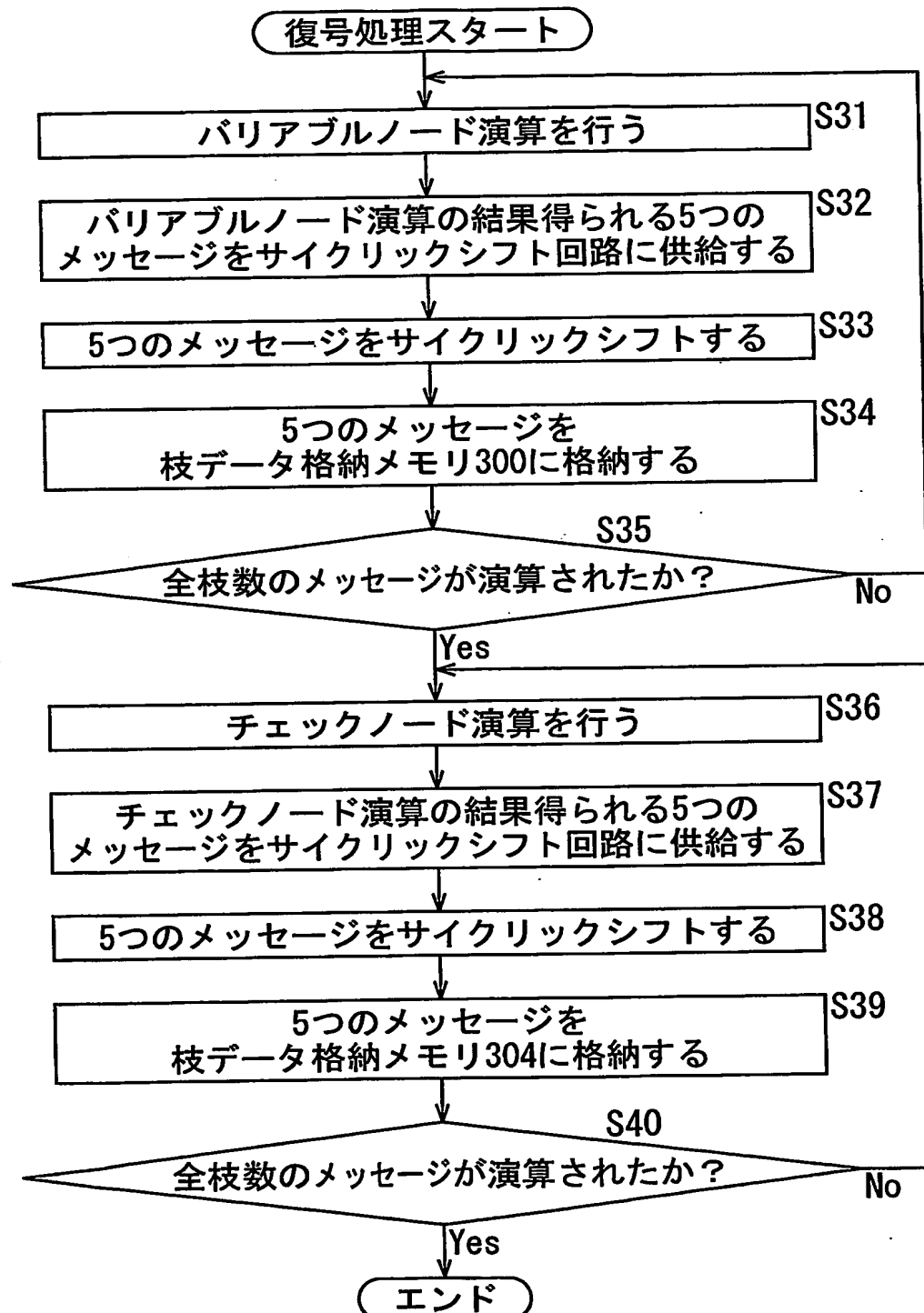






18/35

図17





61 図

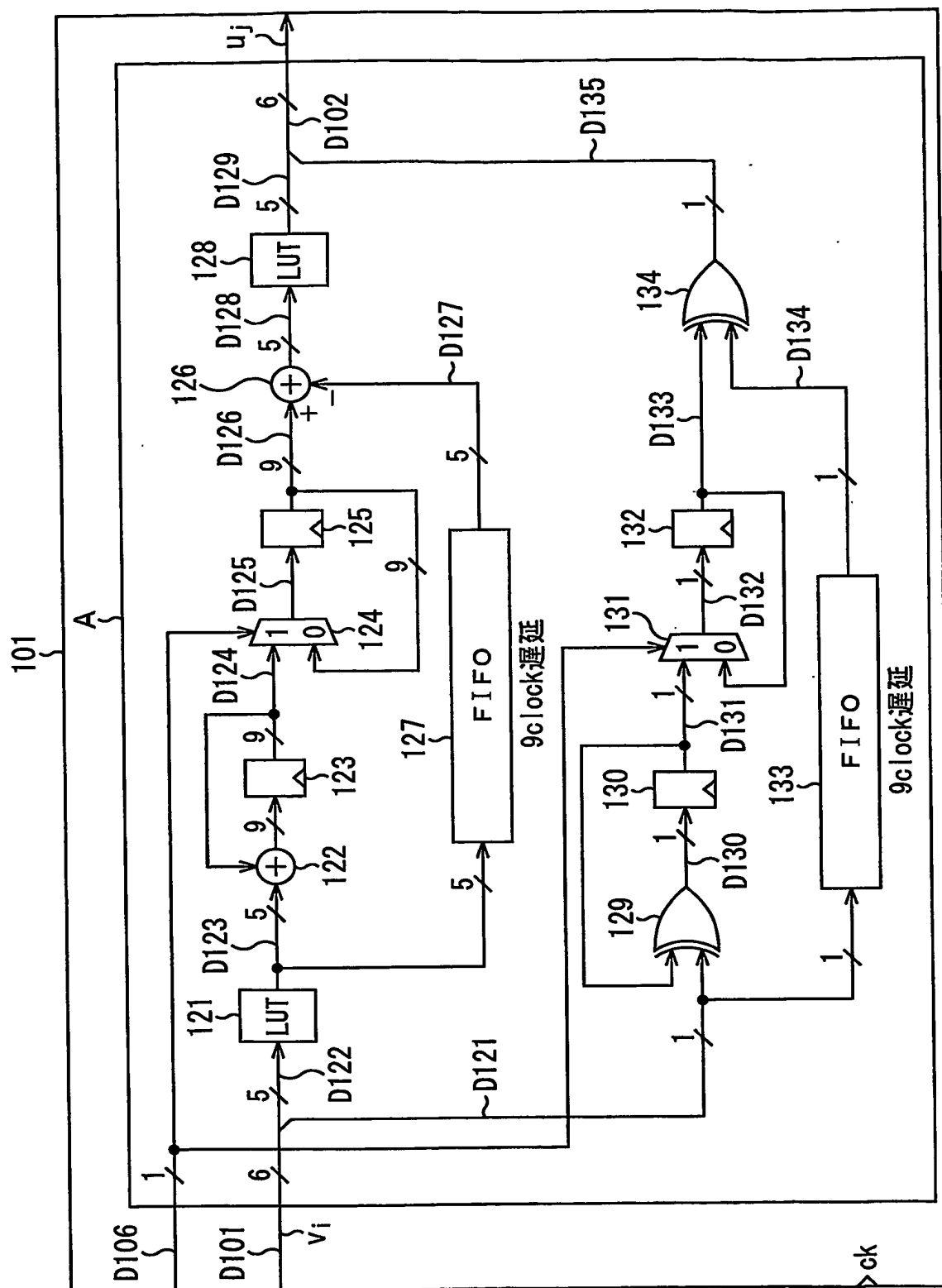
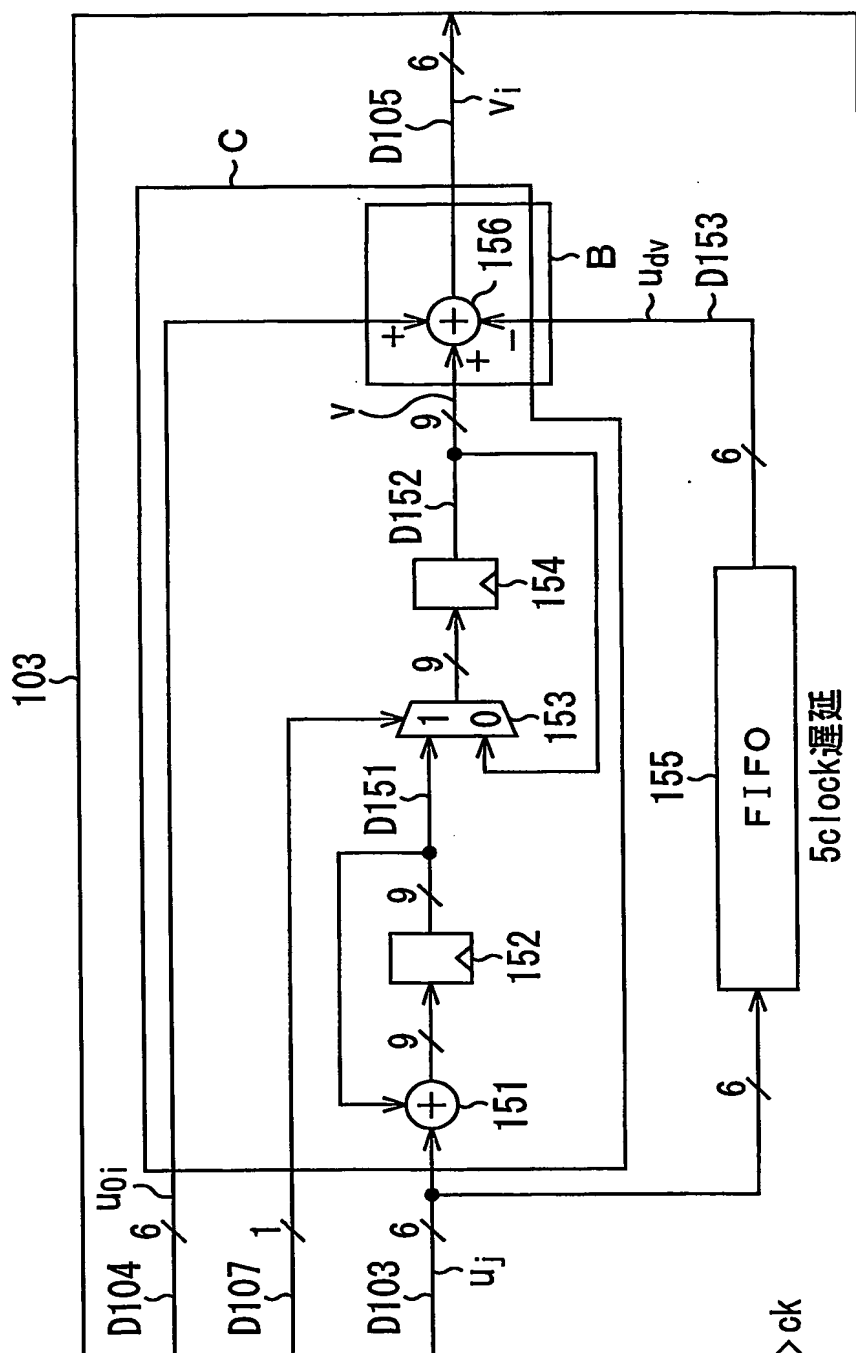


図20



22/35

图21

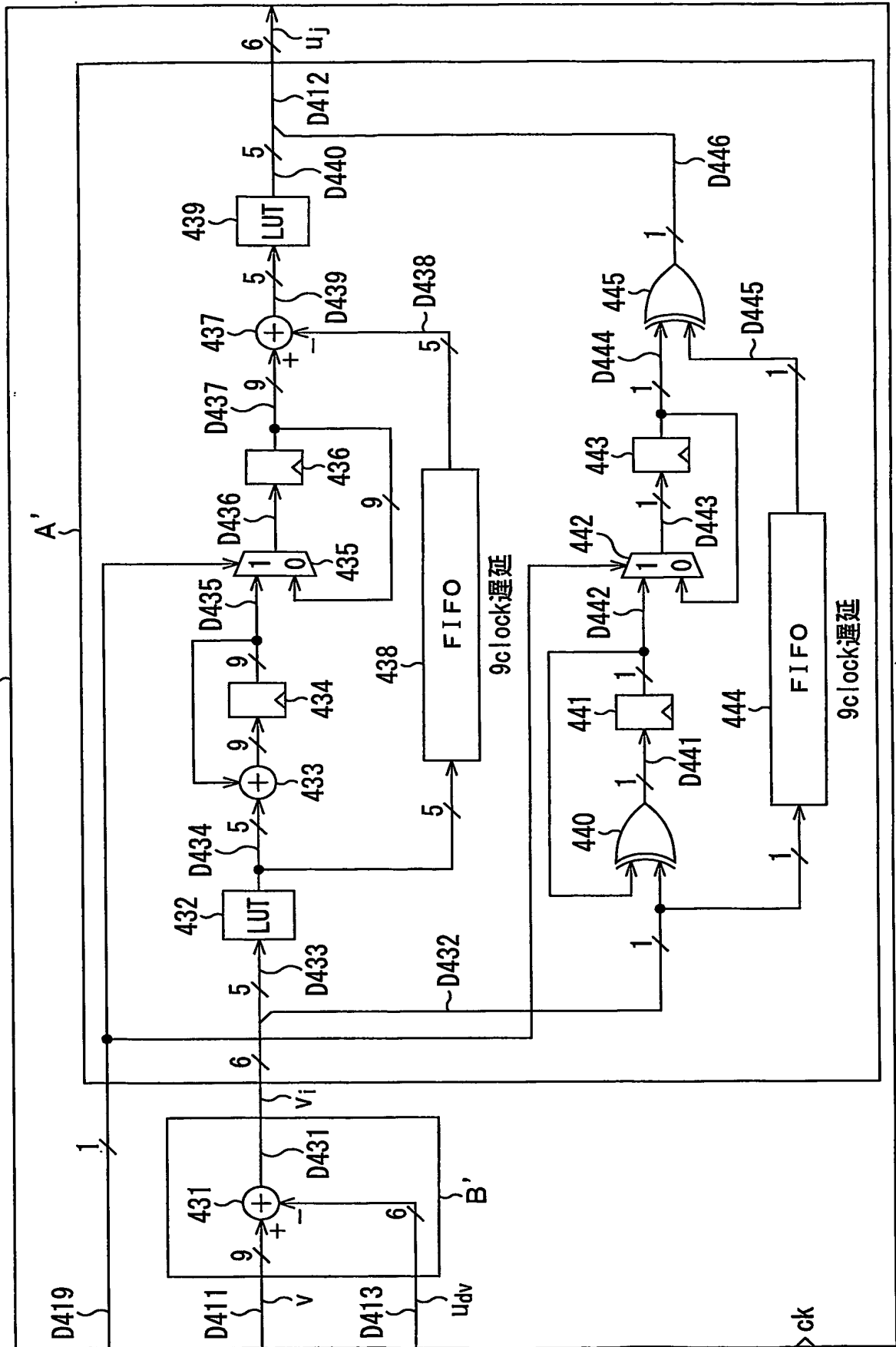
412<sub>1</sub>

図22

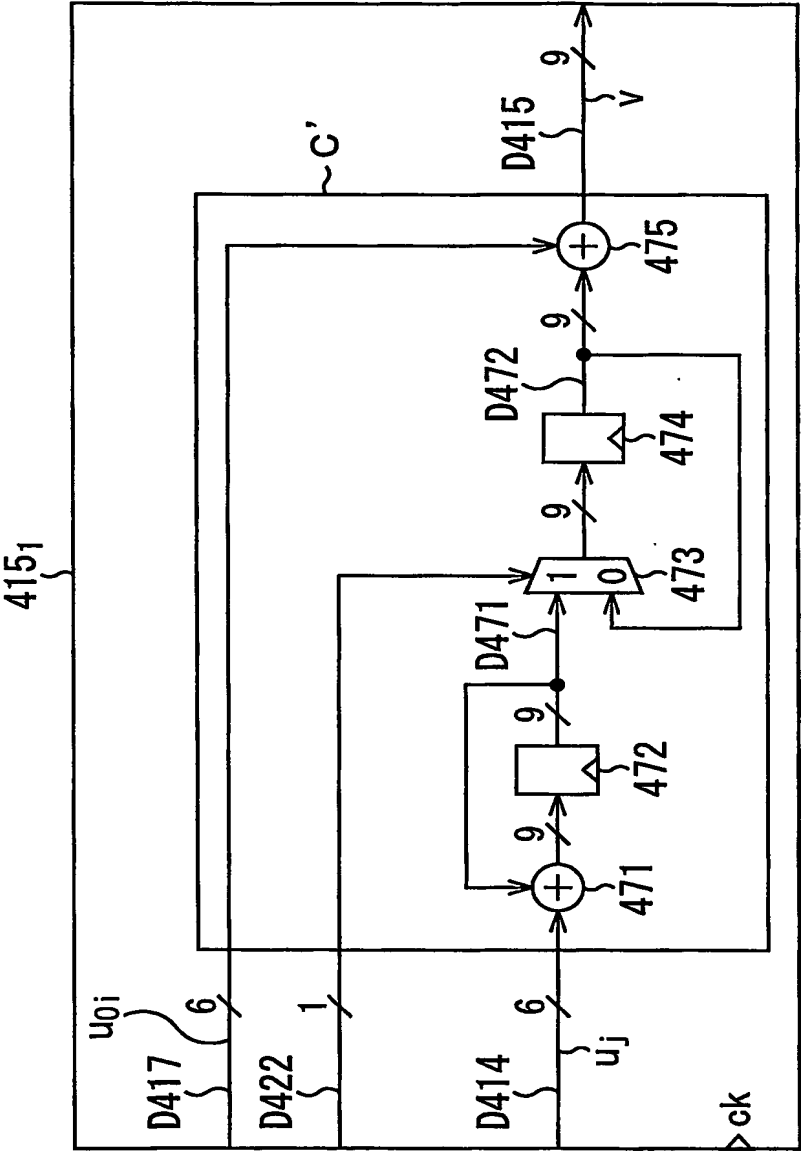


図23

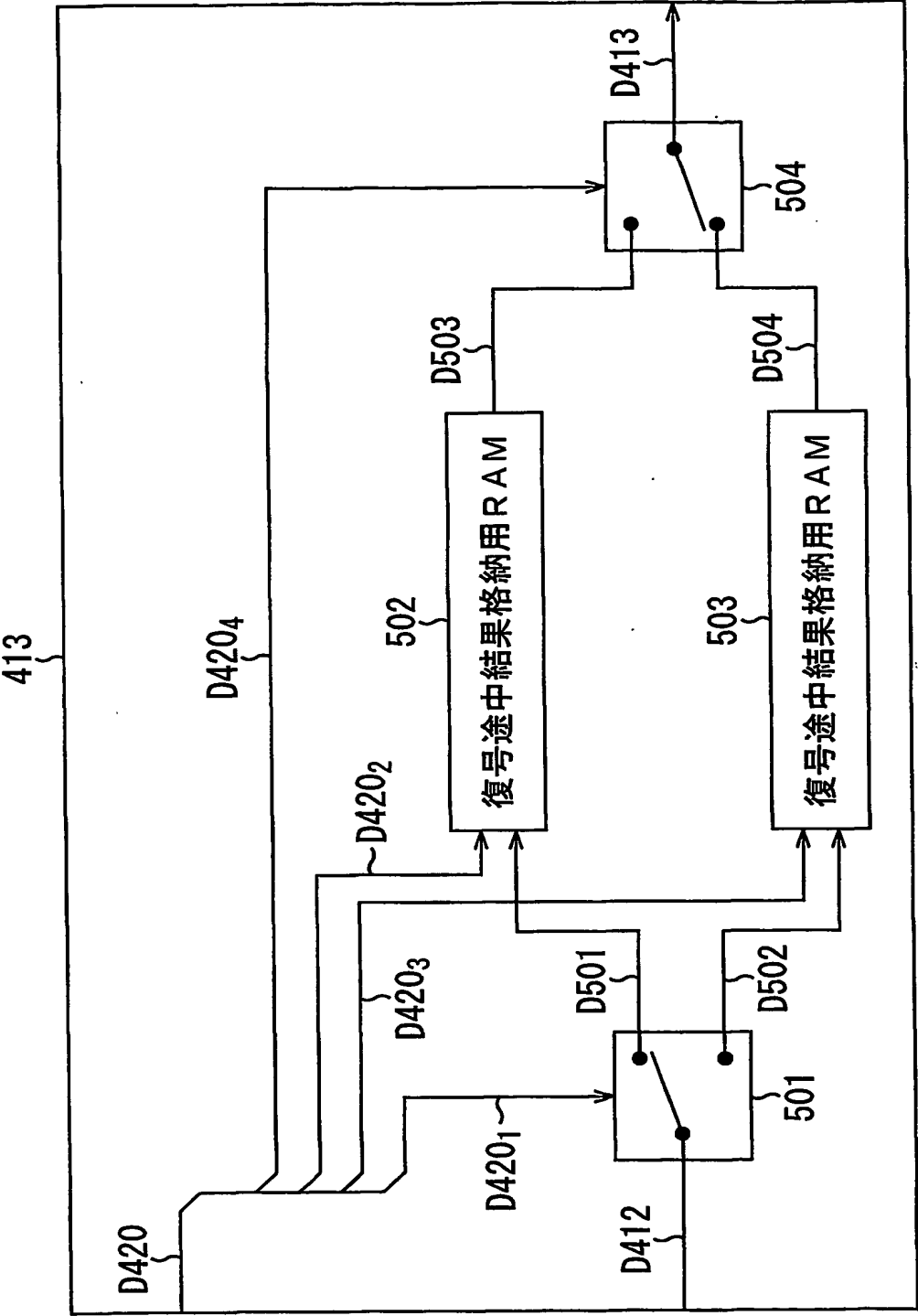
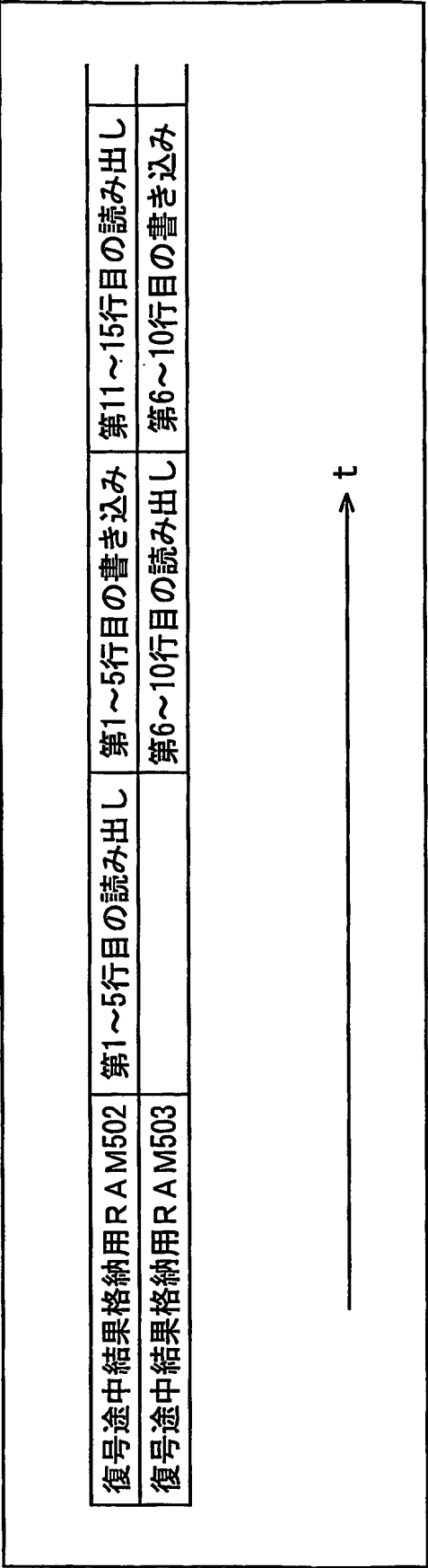




図24



26/35

図25

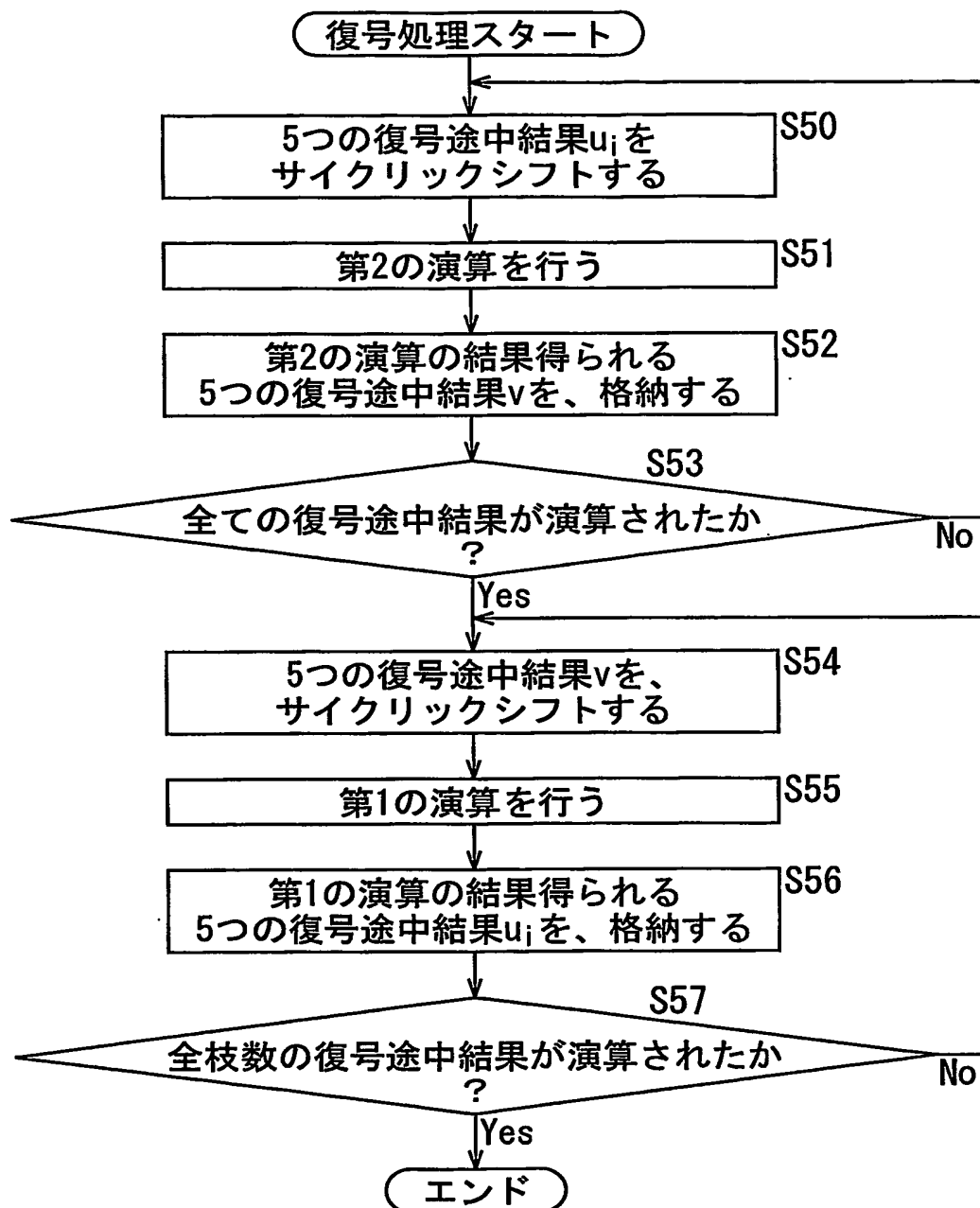


図26

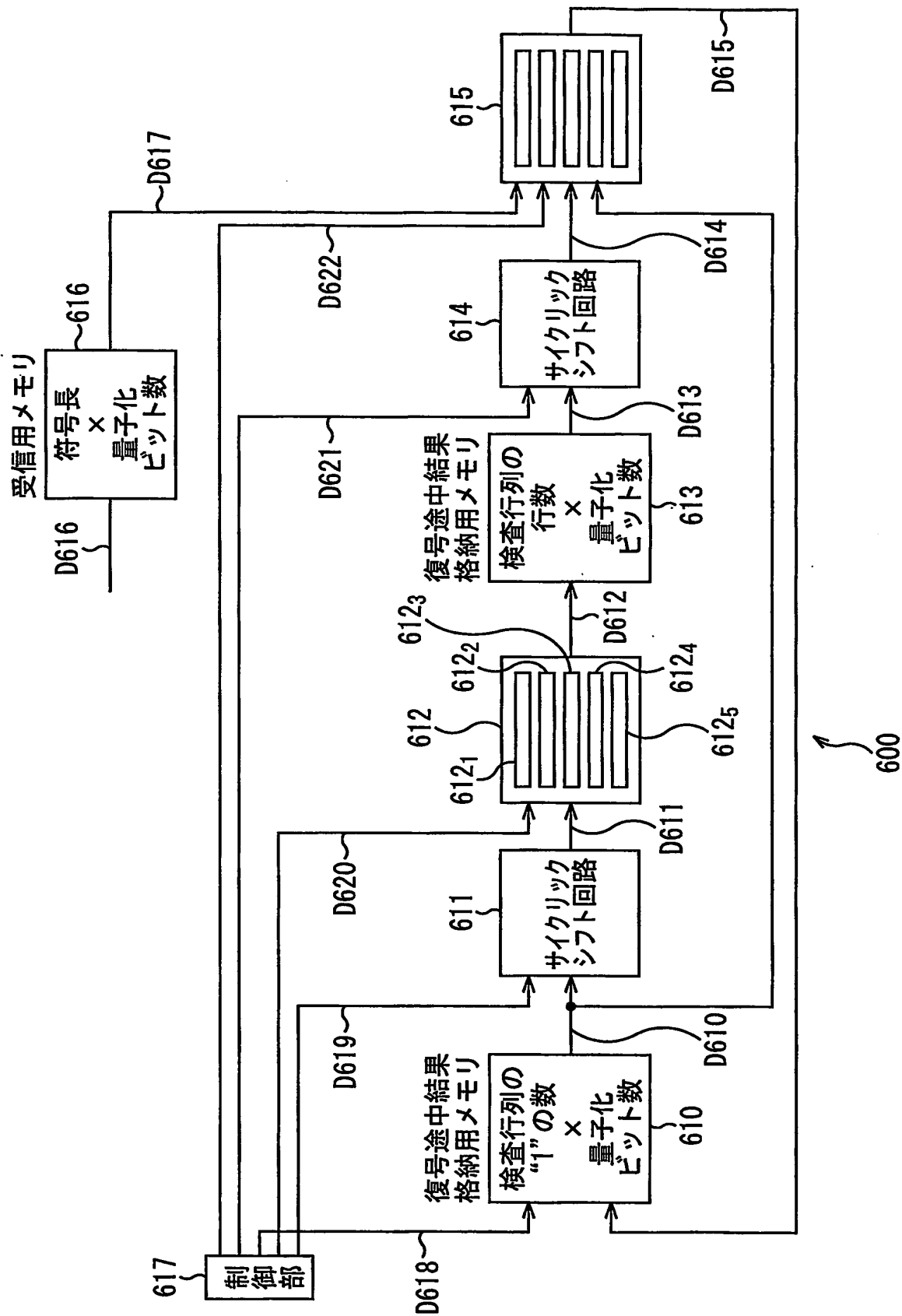


図27

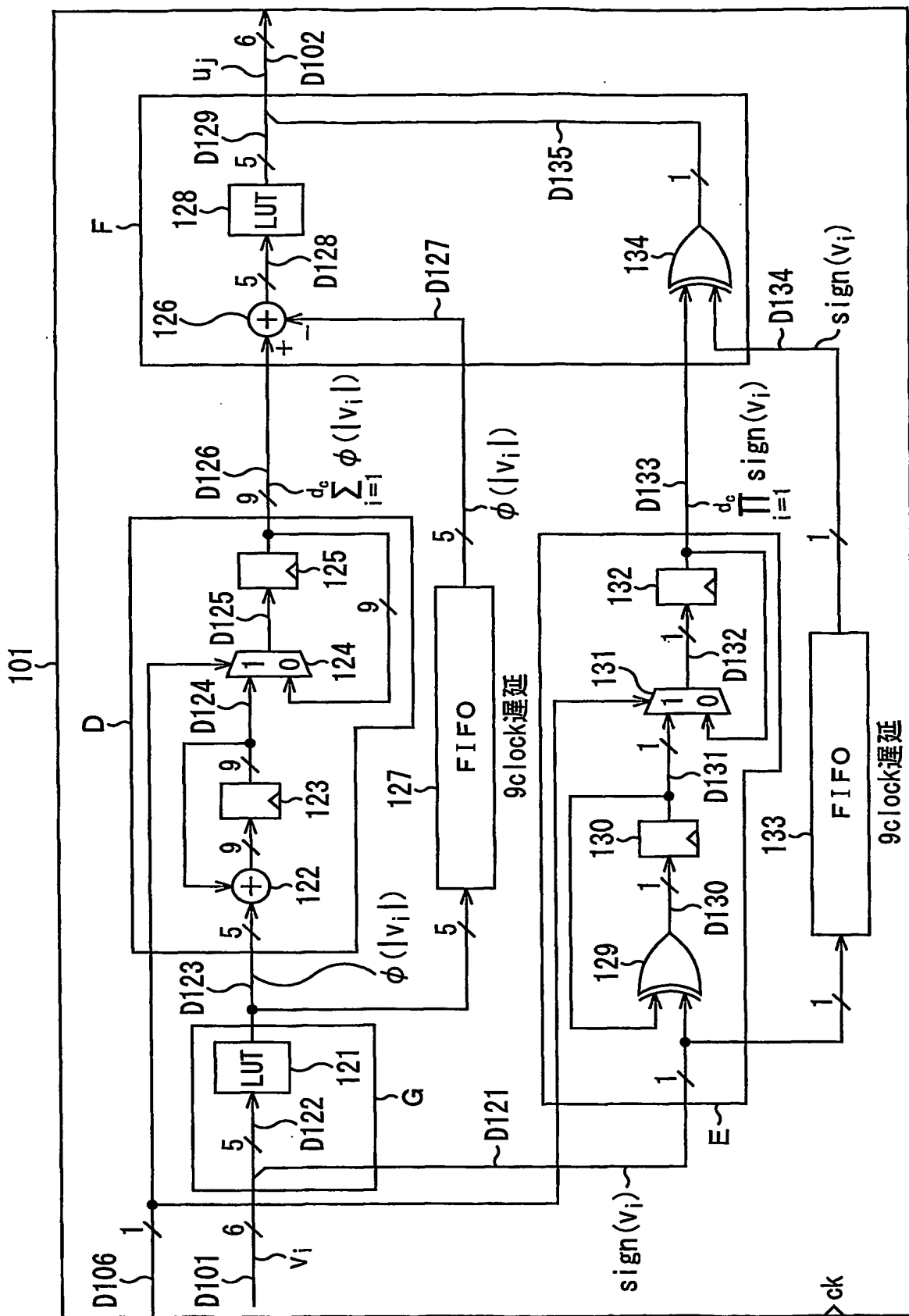


図28

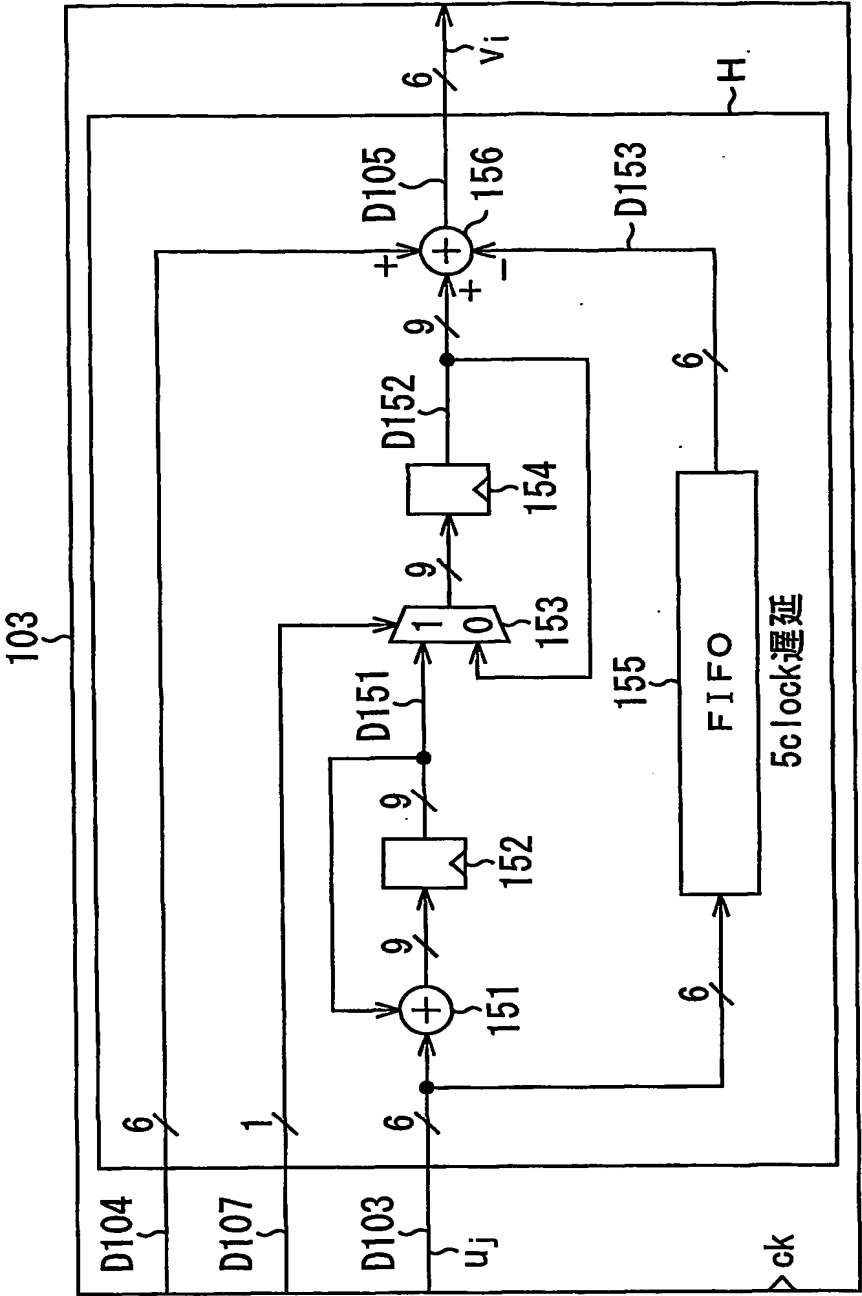
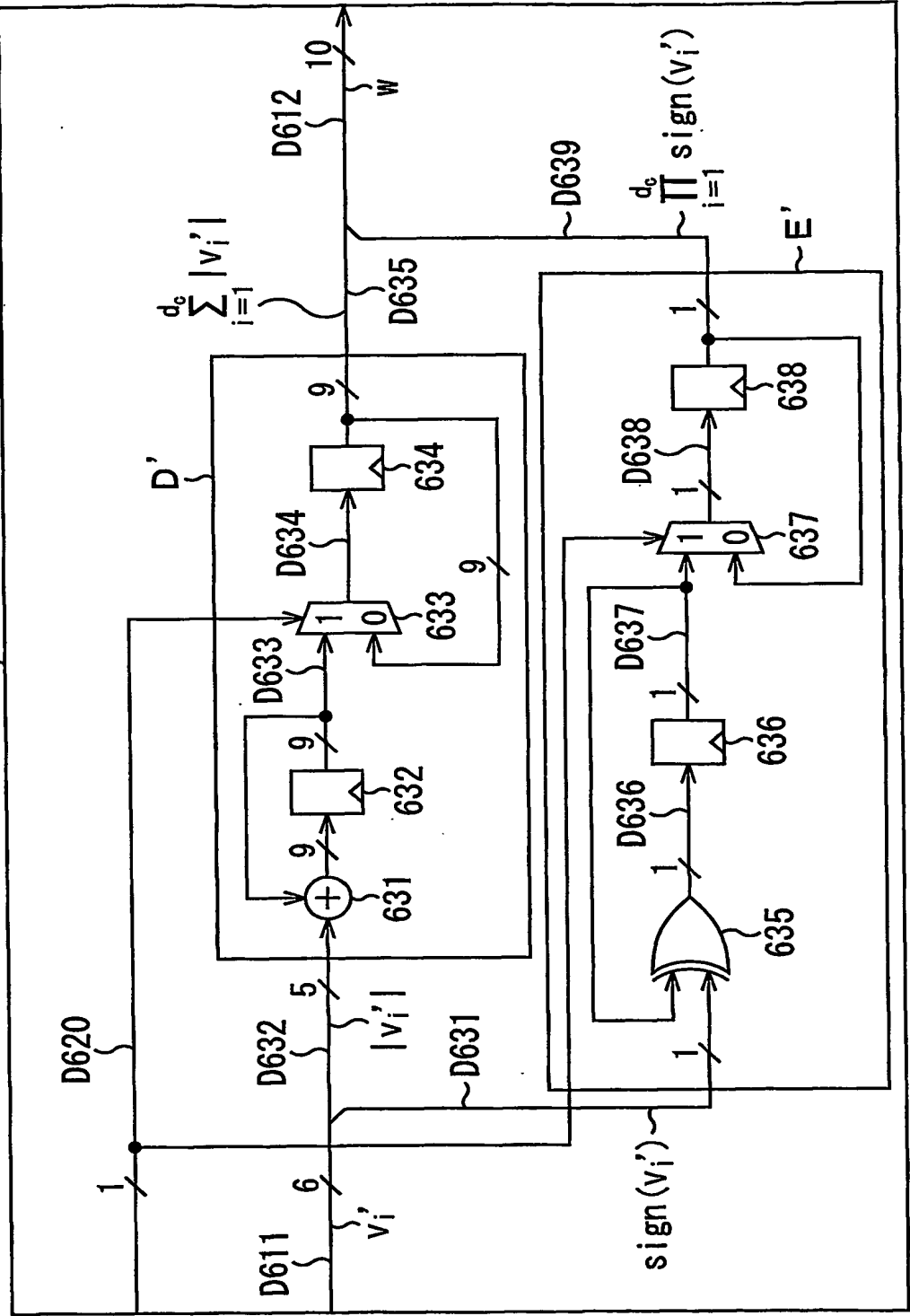


図29

612<sub>1</sub>



31/35

図30

615<sub>1</sub>

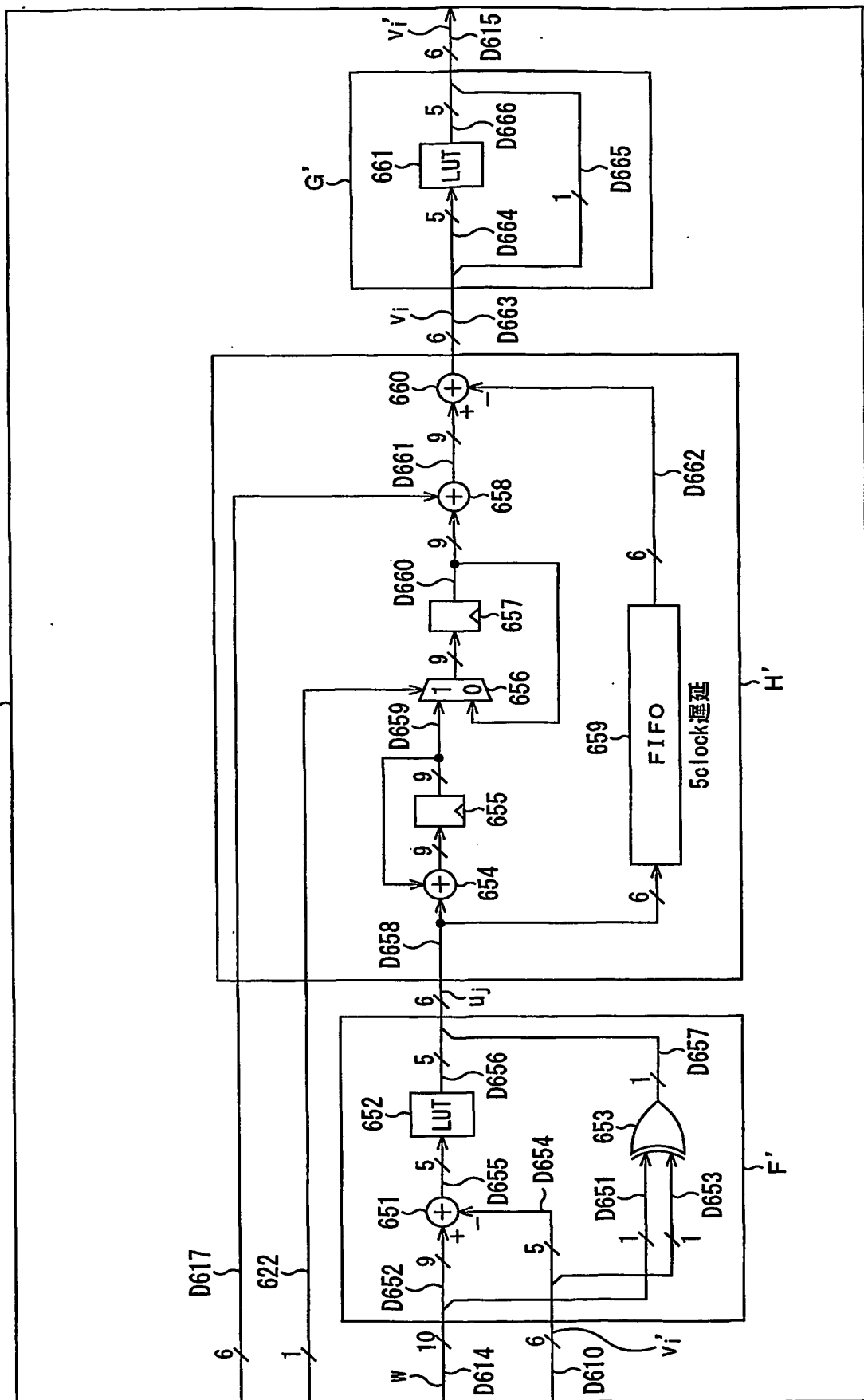


図31

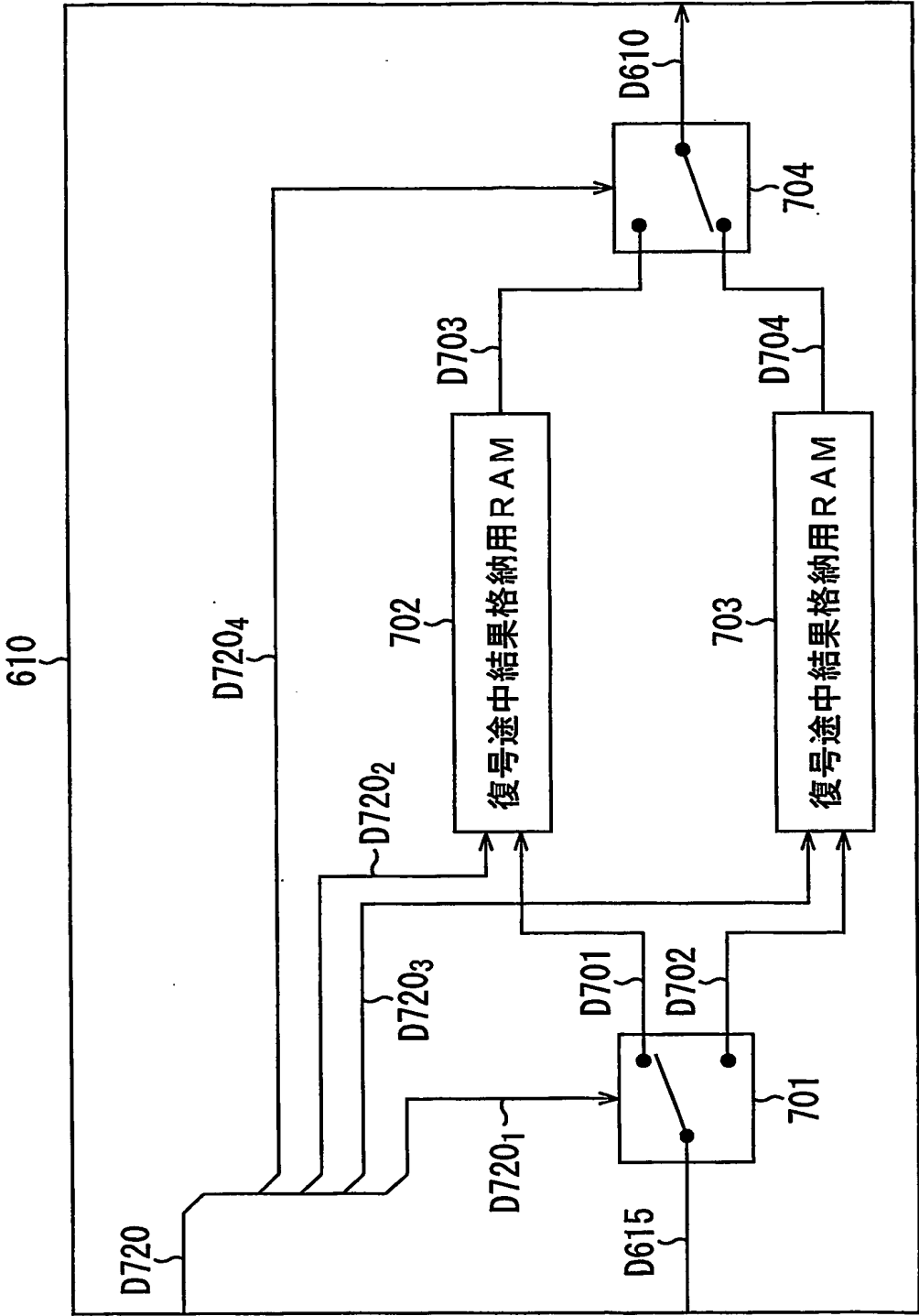
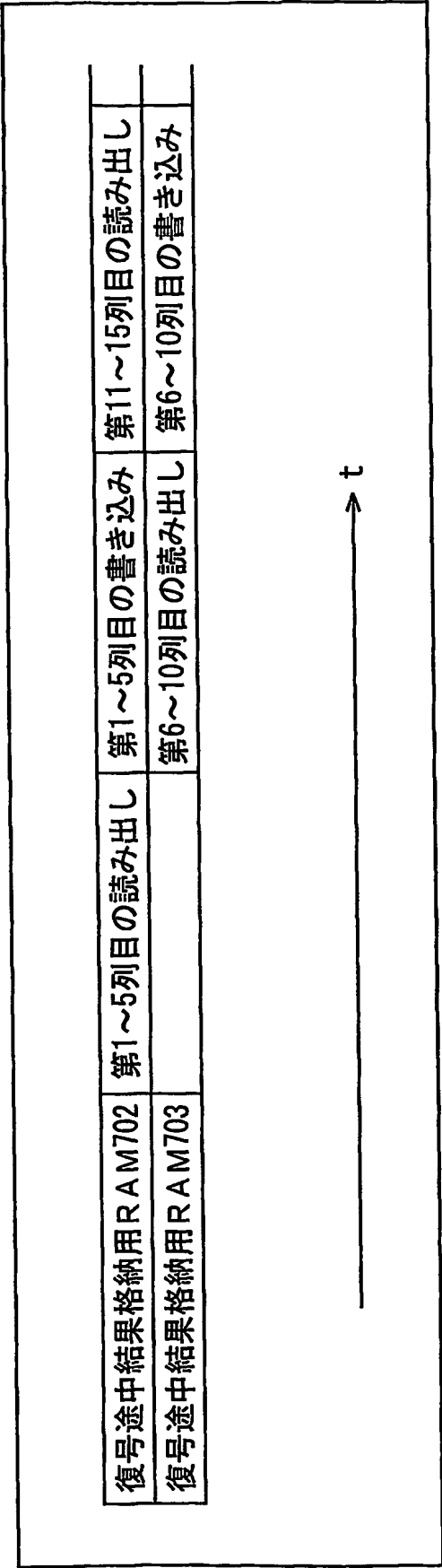


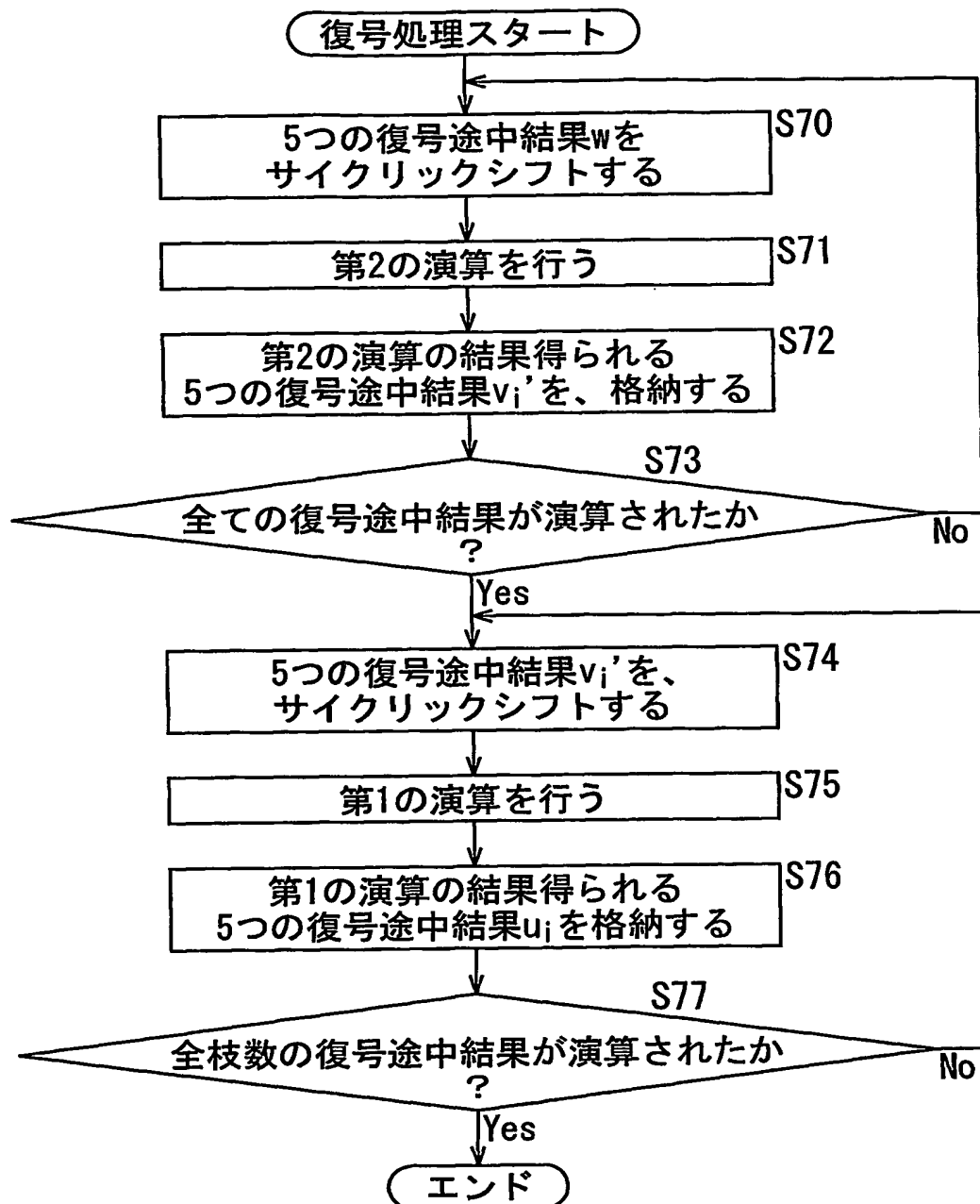


図32



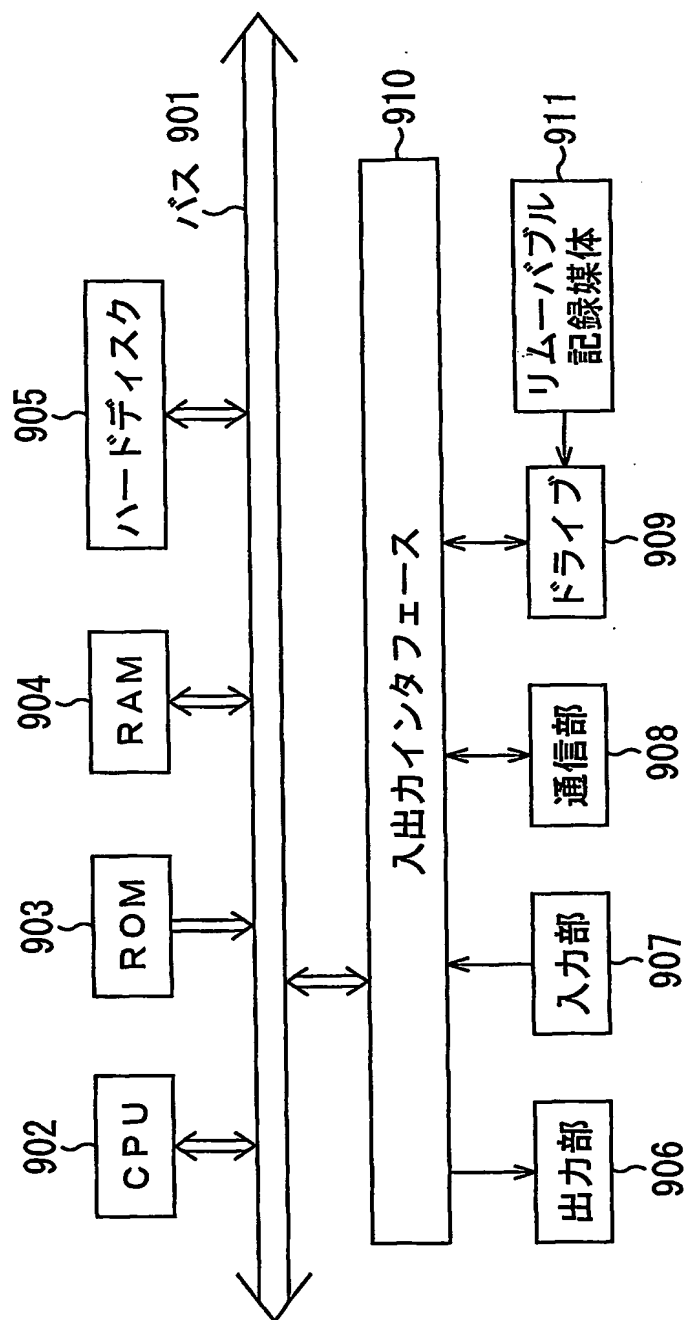
34/35

図33



35/35

図34



# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2004/005562

## A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl<sup>7</sup> H03M13/09, 13/19

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
Int.Cl<sup>7</sup> H03M13/00-13/53

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2004  
Kokai Jitsuyo Shinan Koho 1971-2004 Toroku Jitsuyo Shinan Koho 1994-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
IEEE Xplore (LDPC, Decode, Architecture), IEICE Web (LDPC, TEIMITSUDOPARITY, TEIMITSUDOAYAMARI, FUKUGOU) (in English and in Japanese)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, A	JP 2003-269302 A (Fujitsu Ltd.), 17 October, 2003 (17.10.03), Full text; all drawings (Family: none)	1-41
A	JP 2002-33670 A (Agere Systems Guardian Corp.), 31 January, 2002 (31.01.02), Full text; all drawings & EP 1158682 A2 & US 6539367 B	1-41
A	S. Kim et al., 'Parallel VLSI Architectures for a Class of LDPC Codes', IEEE International Symposium on Circuits and Systems, 2002, ISCAS 2002, Vol.2, No.2002, pages 26 to 29	1-41

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

\* Special categories of cited documents:  
"A" document defining the general state of the art which is not considered to be of particular relevance  
"E" earlier application or patent but published on or after the international filing date  
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)  
"O" document referring to an oral disclosure, use, exhibition or other means  
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention  
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone  
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art  
"&" document member of the same patent family

Date of the actual completion of the international search  
12 July, 2004 (12.07.04)

Date of mailing of the international search report  
27 July, 2004 (27.07.04)

Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

## A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl<sup>7</sup> H03M13/09, 13/19

## B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl<sup>7</sup> H03M13/00-13/53

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年

日本国公開実用新案公報 1971-2004年

日本国実用新案登録公報 1996-2004年

日本国登録実用新案公報 1994-2004年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

IEEE Xplore (LDPC, Decode, Architecture)

IEICE Web (LDPC, 低密度パリティ, 低密度誤り, 復号)

## C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
PA	JP 2003-269302 A (富士通株式会社) 2003. 10. 17, 全文, 全図 (ファミリーなし)	1-41
A	JP 2002-33670 A (アギア システムズ ガーディア ン コーポレーション) 2002. 01. 31, 全文, 全図 & EP 1158682 A2 & US 6539367 B	1-41
A	S. Kim et al, 'Parallel VLSI Architectures for a Class of LD PC Codes', IEEE International Symposium on Circuits and Syst ems, 2002, ISCAS 2002, Volume 2, November 2002, p.26-29	1-41

☐ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

## \* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&amp;」同一パテントファミリー文献

国際調査を完了した日

12. 07. 2004

国際調査報告の発送日

27. 7. 2004

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

田中 庸介

5K

8529

電話番号 03-3581-1101 内線 3555